University of Missouri - Columbia

Master of Science Graduate Thesis

Development of Q-Baller – A Spherical Wheeled Robot

Chapter 4. Basic Control Study of Q-Baller

Jiamin Wang

Jw2gd@mail.missouri.edu

2017/2/27

4. Basic Control Study of Q-Baller

The Dynamics of Q-Baller has been studied in the previous chapter, indicating the feasibility of the current design. Control analysis and controller design relies on the previous results and they will also elevate the dynamics study of Q-Baller to a higher level. In this chapter, we will discuss the stability analysis of Q-Baller and compare the performances of several different controllers designed for the Q-Baller's position and velocity control.

4.1. Fundamentals of Stability Analysis

Achieved from previous dynamic analysis, Q-Baller's system model can be described as below according to (3.21) and (3.22):

$$x = \begin{bmatrix} A & B & C & X & Y & \dot{A} & \dot{B} & \dot{C} & \dot{X} & \dot{Y} \end{bmatrix}^T \tag{4.1}$$

$$u = [U_{x+y+} \quad U_{x+y-} \quad U_{x-y-} \quad U_{x-y+}]^T \tag{4.2}$$

$$v = [T_D^T \quad F_D^T \quad T_d^T \quad F_d^T]^T \tag{4.3}$$

$$\dot{x} = f(x, u, v) \tag{4.4}$$

$$y = g(x, u, w) \tag{4.5}$$

According to Lyapunov Stability Criterion [4-1], if Q-Baller is stable in a certain domain according $x \in D$ to a certain equilibrium point, for an energy function:

$$V = x^T P x \tag{4.6}$$

The system satisfies the following conditions:

1) P is positive definite;

2) In domain D, we have

$$\dot{V} = x^T P \dot{x} + \dot{x}^T P x + x^T \dot{P} x = x^T P f(x, u, v) + f(x, u, v)^T P x + x^T \dot{P} x \leq 0 \tag{4.7}$$

Due to the complexity of Q-Baller System, we have to replace $f(x, u, v)$ with the system's State-Space Equations according to (3.25) and (3.26):

$$\dot{X_{SS}} = A_{SS} X_{SS} + B_{SS} U_{SS} + E_{SS} V_{SS} \tag{4.8}$$

$$Y_{SS} = C_{SS} X_{SS} + D_{SS} U_{SS} + F_{SS} W_{SS} \tag{4.9}$$

Therefore, with the following additional conditions:

1) P is positive definite constant matrix;

2) High order states and external noise are eliminated, and $D_{SS} = 0$;

3) A linear controller is designed for the system:

$$U = -K C_{SS} x \tag{4.10}$$

Function (4.7) can be further simplified to the following state:

$$\dot{V} = x^T (P(A_{SS} - B_{SS} K C_{SS}) + (A_{SS} - B_{SS} K C_{SS})^T P) x \tag{4.11}$$

$$Q = -(P(A_{SS} - B_{SS} K C_{SS}) + (A_{SS} - B_{SS} K C_{SS})^T P) \tag{4.12}$$

From (4.11) and (4.12) it is obvious that to fulfill the requirement of system stability, we must have Q as a symmetric positive definite matrix in domain $x \in D$.

For the convenience purpose, we define the Lyapunov Function to be:

$$\mathcal{L}(A_{\mathcal{L}}, x, Q_{\mathcal{L}}) = P_{\mathcal{L}} \tag{4.13}$$

Here, $Q_{\mathcal{L}}$ is the symmetric positive definite matrix, and $x \in D$ indicating that the Function only works on domain D.

4.2. Linear Controller Design at Zero-Point

As discussed before, the highly nonlinear and state-coupling characteristic if Q-Baller cannot be easily controlled with a linear controller designed at the Zero-Point. The Gain-Scheduling method will help with the problem by applying different controller to the system at different states.

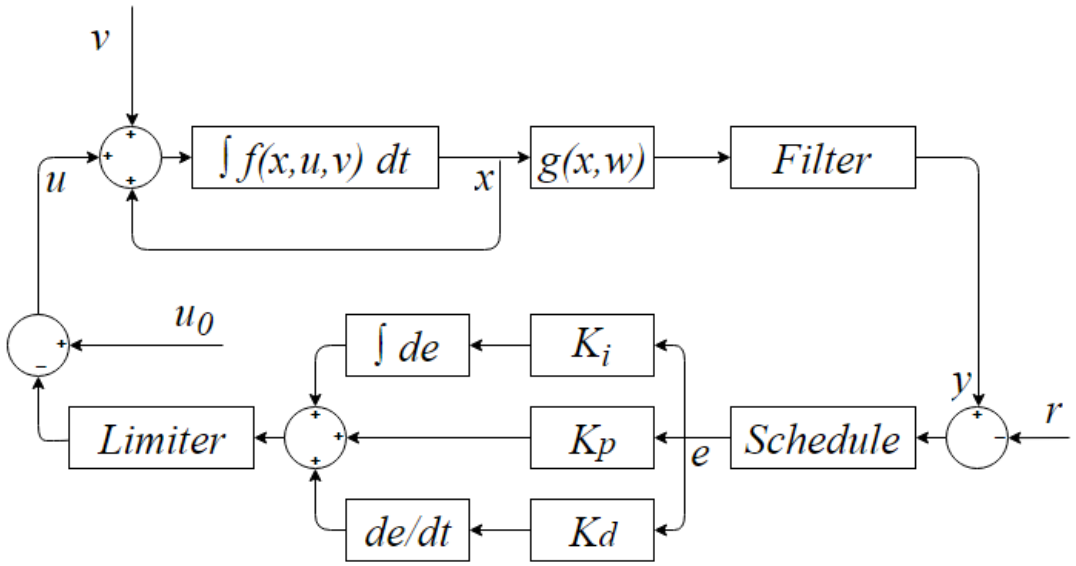


Figure 4.1. Control System Flowchart with Linear Controller

The application of a linear controller to our design is depicted as shown below in Fig. 4.1. For each iteration, the observed states will be filtered and compared with the objective reference. The observed the states will also determine the scheduled controller for input update in this iteration. The error will then be transferred to the controller, updating the input through applying error to the Proportional, Integral and Differential (PID) controller respectively:

$$U_P = K_P e \tag{4.14}$$

$$U_I = K_I \int e(t)dt \tag{4.15}$$

$$U_D = K_D \frac{\partial e(t)}{\partial t} \tag{4.16}$$

In addition to the P Controller which updates the input according to the current state, I Controller tends to eliminate steady state error, and D Controller improves the stability with the tendency of the states.

The Limiter in the Figure 4.1. will limit the input to practical value according the system properties such as input boundaries and maximum input variation limitations.

To adopt the Gain-Scheduling Method, it requires us to design controller at multiple equilibrium states. While the other equilibrium points may be hard to acquire through analytical method due to the complexity of the system, we decide to start from the Zero Point, and then acquire other equilibrium points through experimental means.

Linear Quadratic Regulator (LQR) [4-2] is selected as the original linear controller designer for Q-Baller. LQR controllers are optimal controllers which minimizes the cost function that balances the energy cost and system performance:

$$J = \frac{1}{2}\int_0^\infty x(t)^T Q x(t) + u(t)^T R u(t)\ \ dt \tag{4.17}$$

Here diagonal positive definite matrixes Q and R weighs the performance and energy cost respectively. The controller of LQR is described as:

$$u(t) = -R^{-1}B_{SS}^T P x(t) \tag{4.18}$$

Where P is decided through the matrix algebraic Riccati Equation:

$$-PA_{SS} - A_{SS}^T P - Q + PB_{SS}R^{-1}B_{SS}^T P = 0 \tag{4.19}$$

Through computer tools like MATLAB, the controller can be easily generated. The Controller for the standard model at the Zero-Point is shown as below:

$$K_{ZP} = \begin{bmatrix} 7.381 & -7.399 & -0.500 & -0.316 & -0.316 & 1.035 & -1.046 & -0.272 & -4.333 & -4.333 \\ -7.381 & -7.399 & 0.500 & -0.316 & 0.316 & -1.035 & -1.046 & 0.272 & -4.333 & 4.333 \\ -7.381 & 7.399 & -0.500 & 0.316 & 0.316 & -1.035 & 1.046 & -0.272 & 4.333 & 4.333 \\ 7.381 & 7.399 & 0.500 & 0.316 & -0.316 & 1.035 & 1.046 & 0.272 & 4.333 & -4.333 \end{bmatrix} \tag{4.20}$$

For K_{ZP}, the LQR matrix Q and R are determined as below:

$$Q = diag([100\ \ 100\ \ 50\ \ 20\ \ 20\ \ 50\ \ 50\ \ 25\ \ 10\ \ 10])$$

$$R = diag([50\ \ 50\ \ 50\ \ 50])$$

The stability of Q-Baller is not guaranteed LQR is not guaranteed, but since the controllability of the standard model at Zero-Point we can easily test if there exists a pair of P and Q from (4.12) that satisfies the Lyapunov Stability criterion. After select Q as $I_{10\times10}$, the eigenvalue for the P that satisfies the stability criterion are:

$$eig(P) = [0.015\quad 0.015\quad 0.052\quad 0.320\quad 0.320\quad 1.058\quad 2.477\quad 2.487\quad 9.181\quad 9.211] > 0$$

Since all eigenvalues of P are positive - indicating that P is a positive definite, controller is proved feasible to stabilize the system. Again, the area of stabilization of Q-Baller is not guaranteed due to is nonlinearity.

4.3. Linear Controller Performance Study at Zero Point

Different control system has different characteristics. For the Q-Baller system, the controller generated in the previous subchapter is not perfect. By studying the controller performance, we can conclude to a preliminary control strategy for this system.

As obtained in the previous chapter, K_{ZP} covers 10 states in total, which are the position and velocity states of the 5 basic states. In this case, K_{ZP} is both the PD controller for the position states and the PI controller for the velocity states.

After several preliminary experiments, the following deductions are reached about the Q-Baller system:

1) State A and B are the states that usually does not require velocity control, since they are critical to maintaining the balance of the system. All fast dynamics on A and B should be avoided. A and B also provides the acceleration for X and Y, which is very useful for trajectory tracking and object following. **(D4.1)**

2) PD controllers are preferable for all states during position control, while integral controllers should be implemented to state C, X and Y for velocity control. This is due to the nonlinearity of the system and factors such as viscosity dampers that the static error of velocity controls in most cases are not zero. **(D4.2)**

3) Input limitation should be applied for the system to translate with moderate behavior and avoid abruptions. The limitations include constraining the voltage changes within $12V/s$ and setting the cap for voltage as $\pm 12V$. **(D4.3)**

Deduction 4.1 is an obvious result if the model is analysis with Newtonian method. Deduction 4.2 and 4.3 will be proved below with experimental results of 2 exemplary experiments.

4.3.1. Implementation of PI Controller for Velocity Control

Without I controller, giving that state A and B are still under position control, the controller for the states is presented as below:

$$u = K_{PD\,4\times4}[e_A \quad e_B \quad e_{\dot{A}} \quad e_{\dot{B}}]^T + K_{P\,4\times3}[e_{\dot{C}} \quad e_{\dot{X}} \quad e_{\dot{Y}}]^T + u_r \tag{4.21}$$

Here e is the error between the references and the observed states:

$$e = x_r - x \tag{4.22}$$

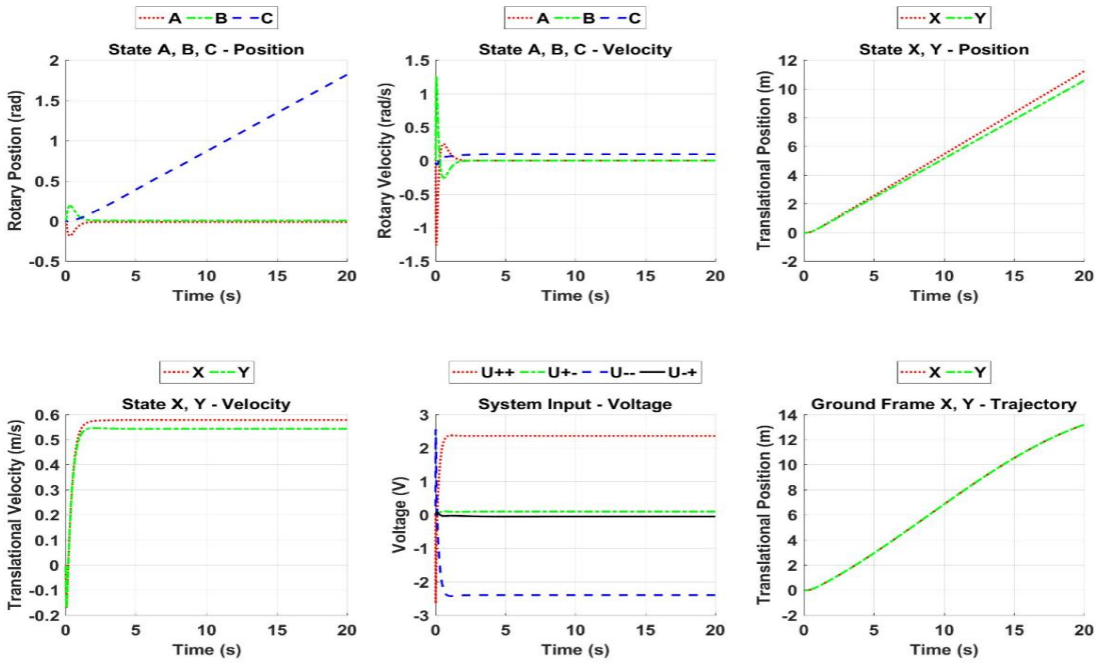
Giving that the objective and the corresponding system input set is $L = [x_r \quad u_r]^T$. If L result in the balance of the system, we shall call L_e an equilibrium objective. Under all equilibrium objectives, the final control outcome will be $e = 0$. This indicates that the system is well-balanced at the objective states.

However, when L is not equilibrium objective, indicating x_{obj} and u_{obj} will not lead to the balance of the system, the result $e = x_{obj} - x \neq 0$. The system may still be statically stable. However, it will converge to another equilibrium point. For an MIMO system like Q-Baller, such problem is common since the system inputs are in most cases affecting multiple states, as shown below in Exp.4.1

Experiment 4.1:

Tracking Reference: $A = 0;\ B = 0;\ \dot{C} = 10\ °/s;\ \dot{X} = 0.3\ m/s;\ \dot{Y} = 0.3\ m/s;$
Noise Simulation: Not Applied;
Observer Simulation: Not Applied;
Input Limitation: Not Applied;
Controller Specification: PD for A, B*; P for* $\dot{C}, \dot{X}, \dot{Y}$*;*

Result:

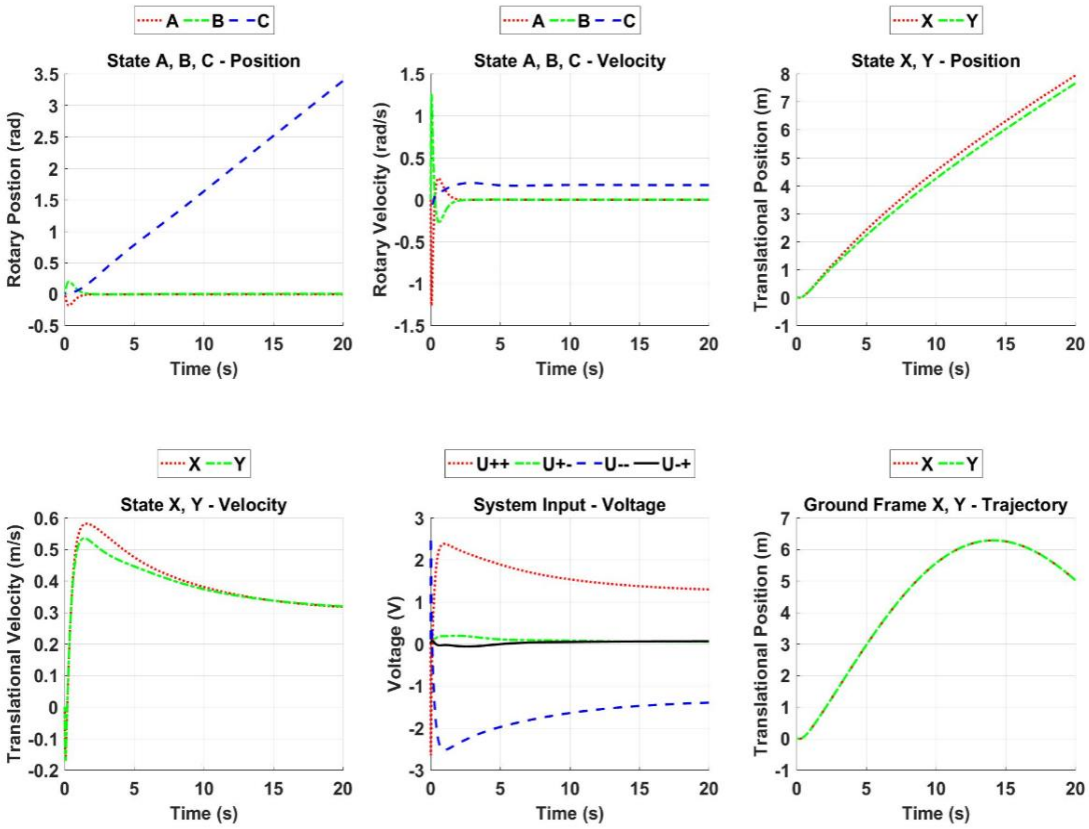


From the result of Exp. 4.1, we can easily realize that the result converged to an equilibrium point which is not the objective. After we changed the control strategy, the effect of PI controller is shown in Exp. 4.2.

Experiment 4.2:

Tracking Reference: $A = 0;\ B = 0;\ \dot{C} = 10\ °/s;\ \dot{X} = \ 0.3\ m/s;\ \dot{Y} = \ 0.3\ m/s;$
Noise Simulation: Not Applied;
Observer Simulation: Not Applied;
Input Limitation: Not Applied;
Controller Specification: PD for A, B*; PI for* $\dot{C}, \dot{X}, \dot{Y}$*;*

Result:



As we can see, the velocity control is accurate this time, which proves the idea in Deduction D4.2. Overshoots in the velocity state plots are expected from the effect of PI controllers. These overshoots may lead to instability of the system.

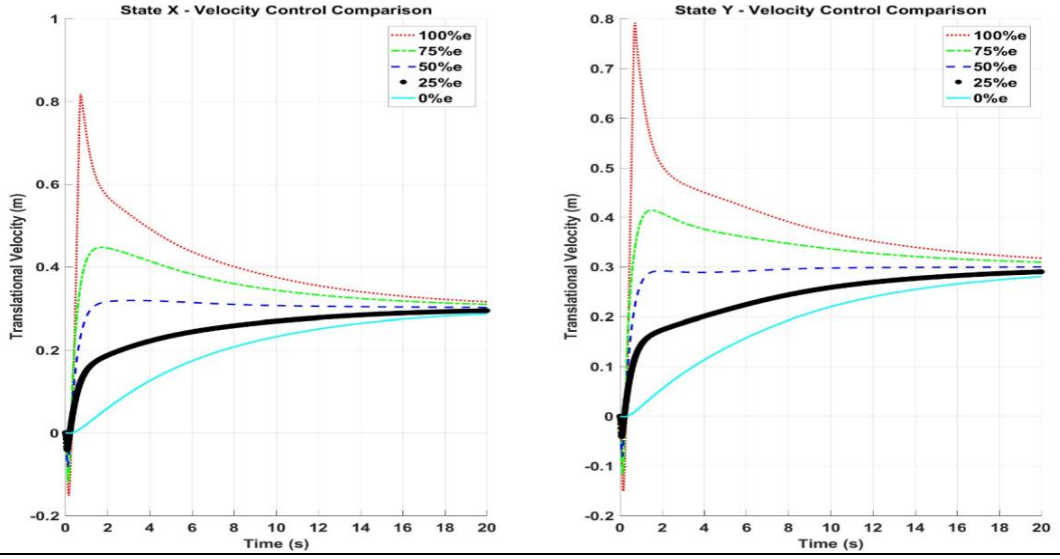
To adjust the overshoots to moderate extents, we proposed a controller named "$P_{0.5}I$" velocity controller especially for X and Y. Through observation, we discovered that K_{ZP} is not tunable due to the stability of the system, while any effective D controller will lead to the same result due to the special characteristic of the system.

We all know that as long as I Controller exists, the system will one day converge to the objective velocity since I controller is actually the position controller of X and Y. For this reason, we attempted to look at the velocity control error as a dynamic actuation of the velocity controller. We did not change K_{ZP} but instead experimented different velocity error for the velocity control, as shown in Exp. 4.3. The velocity error given are 100%, 75%, 50%, 25% and 0% (which is equivalent to position control with time varying objective) of the actual velocity error, respectively. The experiment objective is the same as Exp.1 and Exp.2, but this time, we will compare the velocity state of the outcomes.

Experiment 4.3:

Tracking Reference: $A = 0;\ B = 0;\ \dot{C} = 10\ °/s;\ \dot{X} = \ 0.3\ m/s;\ \dot{Y} = \ 0.3\ m/s;$
Noise Simulation: Not Applied;
Observer Simulation: Not Applied;
Input Limitation: Not Applied;
Controller Specification: PD for A, B*; Compared PI for* $\dot{C}, \dot{X}, \dot{Y}$ *with different initial velocity error;*

Result:



From the result of Exp. 4.3, it is very obvious that the 50%e plot have the most satisfying result for velocity control. Based on such observation, we applied the so called "$P_{0.5}I$" velocity controller to X and Y. However, it should be noted that the technique may be excellent for velocity control, but it may not be the best method for position control of X and Y.

4.3.2. Implementation of Input Limiter

Input Limiter is more related to the practical property of the system. In chapter 3, we introduced the dynamics of the brushed DC motors:

$$\frac{T_{s_0} U}{U_0} - T_m = J_w \dot{\omega} + b_w \omega + \frac{60 T_{s_0}}{2\pi n_{0_0}} \omega \tag{4.23}$$

Equation (4.23) is a function that omitted the motor's fast dynamics which are related to the inner electric circuitry of the DC motor. In real life the output of the motor is affected by these factors, which will result in a slight lag in motor response. The maximum output of the motor is also affected by the maximum voltage power of the motor.

An algorithm is designed for the operation of the motors named as "Input Limiter". Input limiter will be used in both simulation and real practice. While in real world application input limiter may also contain features such as overcoming motor cogging torque and voltage initial deviations. Here in simulation the algorithm only contains the two features described above:

1) $If\left(|u(t) - u(t-1)| > \frac{h * u_{max}}{t_{rise}}\right) \gg u(t) = u(t-1) + \frac{u(t)-u(t-1)}{|u(t)-u(t-1)|} * \frac{h * u_{max}}{t_{rise}}$ **(Alg.4.1.1)**

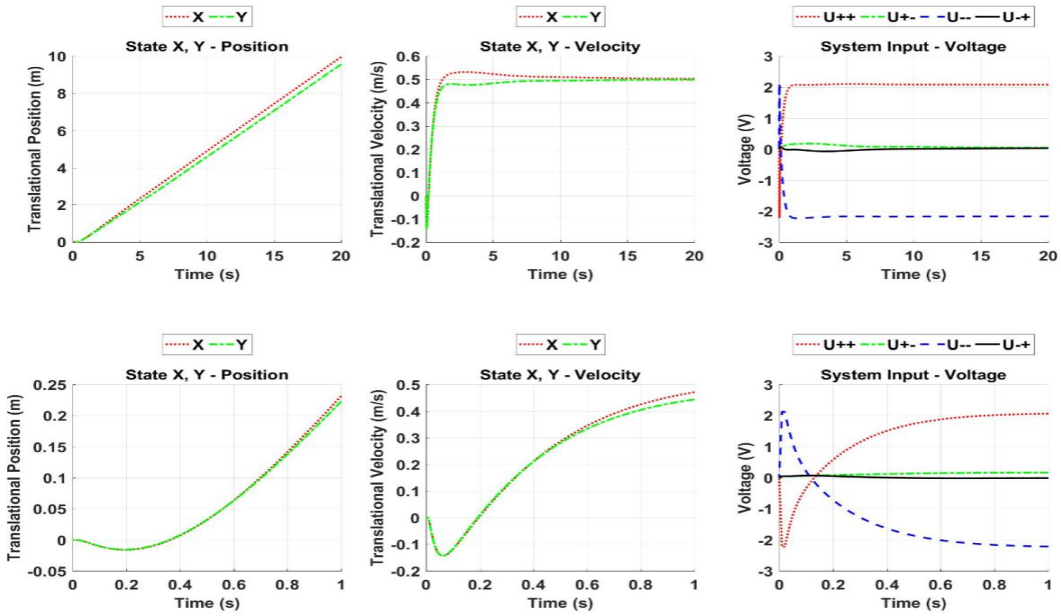
2) $If(|u(t)| > u_{max}) \gg u(t) = \frac{u(t)}{|u(t)|} * u_{max}$ **(Alg.4.1.2)**

Here, t_{rise} is the required time for voltage to rise from 0V to u_{max}, and h is the time difference between t and $t-1$. The effect of Input Limiter is demonstrated in Exp. 4.4:

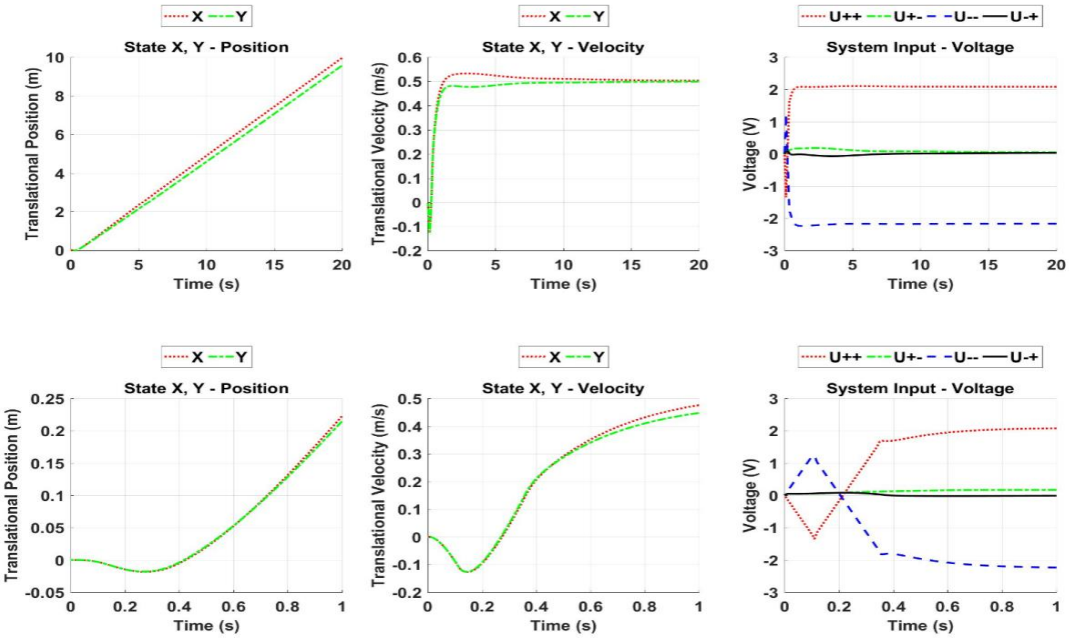
Experiment 4.4:

Tracking Reference: $A = 0;\ B = 0;\ \dot{C} = 10\ °/s;\ \dot{X} = \ 0.3\ m/s;\ \dot{Y} = \ 0.3\ m/s;$
Noise Simulation: Not Applied;
Observer Simulation: Not Applied;
Input Limitation: Not Applied Vs. (Power Rise time = 1 s; Maximum Voltage = 11.1 V;)
Controller Specification: PD for A, B*;* $P_{0.5}I$ *for* $\dot{C}, \dot{X}, \dot{Y}$*;*

Result.1 (Without Input Limitation):



Result.2 (Applied Input Limitation):



As you can see, the performance of the motor is hardly affected by the limiter when you look at the long term performance, but the dynamics in the first second has been affected drastically. With input limiter, the system is obviously operating in a more moderate behavior, proving the idea in Deduction D4.3. However, it should be stated that the input limiter will in some ways affect the robustness of the controller, which is why its parameter should be selected wisely.

4.3.3. Modification of Controller for improved controller Performance

During the experiments, we also discovered a disappointment in the Yawing response quickness due to the weak controller gain according to the Yaw states. The settling time for the Yaw state would not satisfy the performance requirement. To solve this problem, we modified the parameter of the LQR controller to generate a new linear controller according to zero point:

$$Q_{New} = diag([100\ \ 100\ \ 10000\ \ 20\ \ 20\ \ 50\ \ 50\ \ 5000\ \ 10\ \ 10])$$

$$R_{New} = diag([50\ \ 50\ \ 50\ \ 50])$$

Even when the parameters for the yawing look exaggerated, the new LQR Controller is generated to be:

$$K_{New} = \begin{bmatrix} 7.381 & -7.399 & -7.0711 & -0.316 & -0.316 & 1.035 & -1.046 & -4.8897 & -4.333 & -4.333 \\ -7.381 & -7.399 & 7.0711 & -0.316 & 0.316 & -1.035 & -1.046 & 4.8897 & -4.333 & 4.333 \\ -7.381 & 7.399 & -7.0711 & 0.316 & 0.316 & -1.035 & 1.046 & -4.8897 & 4.333 & 4.333 \\ 7.381 & 7.399 & 7.0711 & 0.316 & -0.316 & 1.035 & 1.046 & 4.8897 & 4.333 & -4.333 \end{bmatrix} \quad (4.24)$$

From the new controller we can easily recognize that the controller for states other than C and $\dot{C}$ are not changed, indicating that the yawing state are irrelevant to the other states when the system is linearized at the Zero Point. The controller matrix elements on the 3rd and the 8th column of the matrixes are magnified by $10\sqrt{2}$ as a result of the corresponding parameter in the Q magnified by 200 times. The new controller also satisfies the Lyapunov Stability Criterion:

$$eig\left(\mathcal{L}\left((A_{ZP} - B_{ZP}K_{New}), x, I_{10\times 10}\right)\right) = eig(P_n)$$

$$eig(P_n) = [0.004\quad 0.014\quad 0.014\quad 0.529\quad 0.531\quad 1.0605\quad 5.048\quad 5.059\quad 6.400\quad 6.429] > 0$$

Exp. 4.5 compares the performance of the old and new controllers. The new controller responses much faster by applying a much bigger input into the system. The new controller also alleviated the overshooting problem which existed in the old system. The overall performance of the controller is improved. However, we need to be careful with the error as a result of an increase in controller power.

4.4. Gain Scheduled Controller Design

The linear controller at Zero Point is proposed in Chapter 4.2 and improved in Chapter 4.3. However, this controller is very limited as it is designed based on a linearized system. Gain scheduled controller, as mentioned before, extend the use of basic linear control method to satisfy the requirement of stabilizing a nonlinear system [4-1].

4.4.1. Fundamentals of Gain Scheduling

For a MIMO system without exterior disturbances:

$$\dot{x} = f(x, u) \quad (4.25)$$

$$y = g(x, u) \quad (4.26)$$

Assuming that the system has the following properties:

1) Continuous and differentiable in the domain $\{x, u\} \in D_{GS}$;

2) Has multiple equilibrium points $\{x_e, u_e\}$ in domain D_{GS} that satisfies:

$$\dot{x}_e = f(x_e, u_e) = 0 \quad (4.25)$$

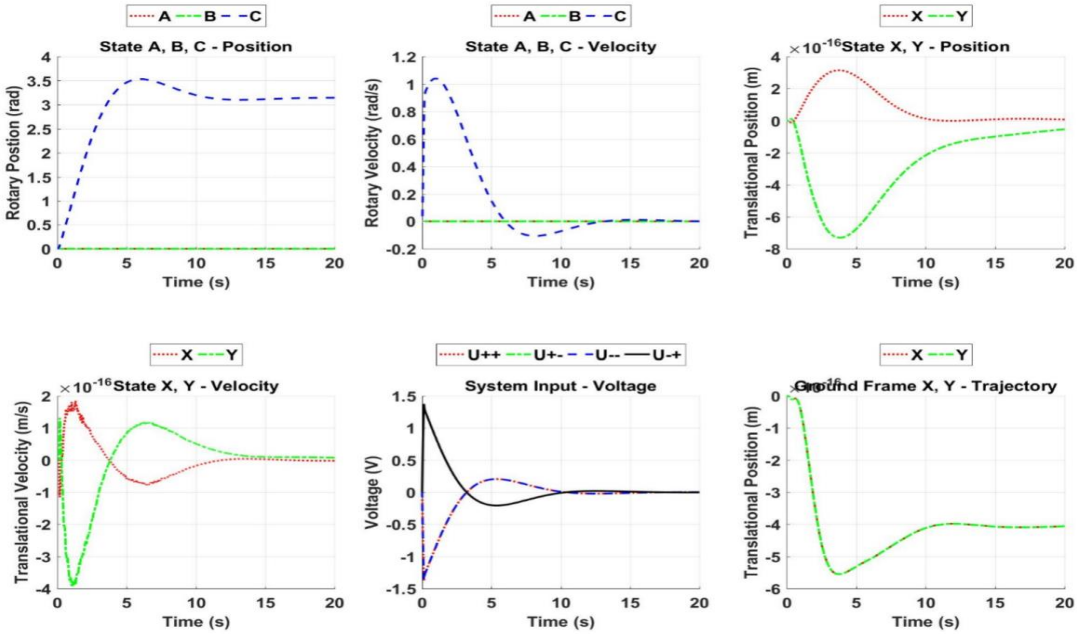
Then the system can be linearized at the equilibrium points as:

$$\dot{X_{SS}} = \dot{X_{SS}} - \dot{X}_e = A_{SS}(x_e, u_e)(X_{SS} - X_e) + B_{SS}(x_e, u_e)(U_{SS} - U_e) \quad (4.26)$$

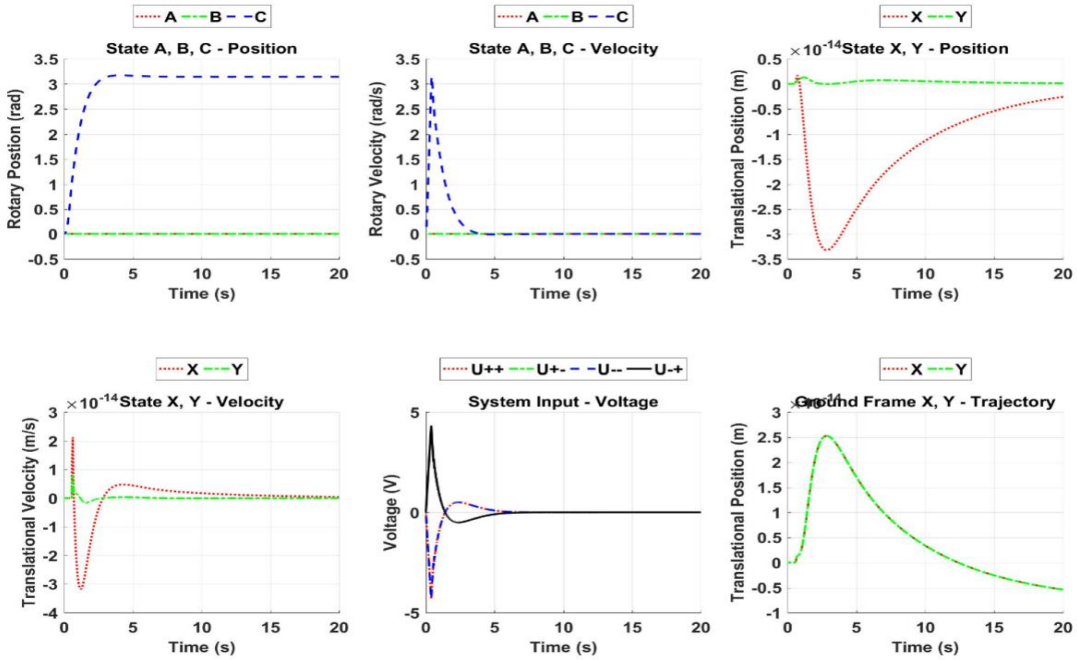
Experiment 4.5:

Tracking Reference: $A = 0;\ \ B = 0;\ \ C = \pi;\ \ X = 0;\ \ Y = 0;$
Noise Simulation: Not Applied;
Observer Simulation: Not Applied;
Input Limitation: Power Rise time = 1 s; Maximum Voltage = 11.1 V;
Controller Specification: Old PD Controller Vs. New PD Controller for $A, B, C, X, Y;$

Result.1 (Old Controller):



Result.2 (New Controller):



Giving that a feasible fixed linear controller K_S has been designed at $\{x_e, u_e\}$, the control input can be described as:

$$U_{SS} = -K(x_e, u_e)(X_{SS} - X_e) + U_e \tag{4.27}$$

The system linearized at the equilibrium point can thus be rewritten as:

$$\dot{X_{SS}} = (A_{SS}(x_e, u_e) - B_{SS}(x_e, u_e)K(x_e, u_e))(X_{SS} - X_e) \tag{4.28}$$

For different tracking reference x_r, the linearized system can be rewritten as

$$\dot{X_{SS}} = (A_{SS}(x_r, u_r) - B_{SS}(x_r, u_r)K(x_r, u_r))(X_{SS} - X_r) \tag{4.29}$$

In conclusion, the closed-loop nonlinear system can be described as:

$$\dot{x} = f(x, -K(x_r, u_r)(x - x_r) + u_r) \tag{4.30}$$

$$y = g(x, -K(x_r, u_r)(x - x_r) + u_r) \tag{4.31}$$

4.4.2. The Traditional Gain Scheduling

The first thing for gain-scheduling is to design the operating points. For our Q-Baller, there are 10 states in total. However, state C, X and Y are irrelevant to the State-Space Matrix A_{SS} and B_{SS}. Of the remaining 7 states, considering the relatively higher robustness of state $\dot{A}$, $\dot{B}$ and $\dot{C}$, we focus the scheduling of the gain according to $\dot{X}$ and $\dot{Y}$ to design a more power velocity controller.

The equilibrium points are acquired through simulation experiments since it is very difficult to calculate those points directly through the governing ODEs of the system. Starting from the Zero Point, the system is controlled to accelerate to and maintaining balance at certain velocity.

Since K_{ZP} is not robust enough to maintain balance when the combined velocity of $\dot{X}$ and $\dot{Y}$ go beyond $v = 1\ m/s$, the gain-scheduled controller was first designed at several low speed equilibrium points. The system then used the preliminary gain-scheduled controller to reach higher speeds and thus gradually expand the range of the controller. The final gain scheduling point design is presented below in Fig. 4.2.

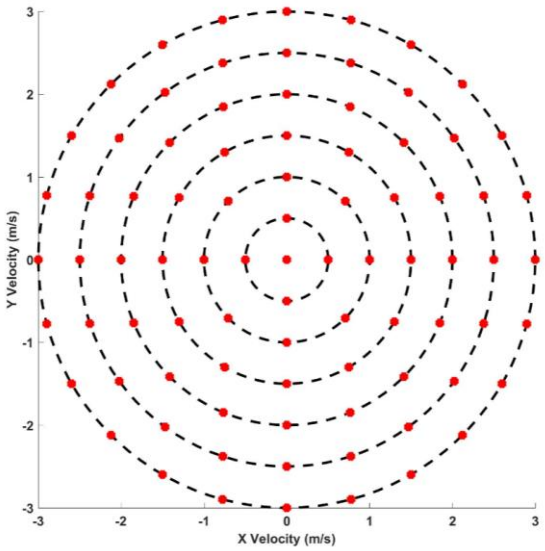


Figure 4.2. The Equilibrium Points for Gain Scheduling

There are totally 85 gain scheduling operating points, including the Zero Point, 4 points at $v = 0.5\ m/s$, 8 points at $v = 1\ m/s$, 12 points at $v = 1.5\ m/s$, 16 points at $v = 2\ m/s$, 20 points at $v = 2.5\ m/s$ and 20 points at $v = 3\ m/s$. LQR Controllers are designed at the equilibrium points with the parameters:

$$Q = diag([100\ \ 100\ \ 5000\ \ 20\ \ 20\ \ 50\ \ 50\ \ 3000\ \ 10\ \ 10])$$

$$R = diag([50\ \ 50\ \ 50\ \ 50])$$

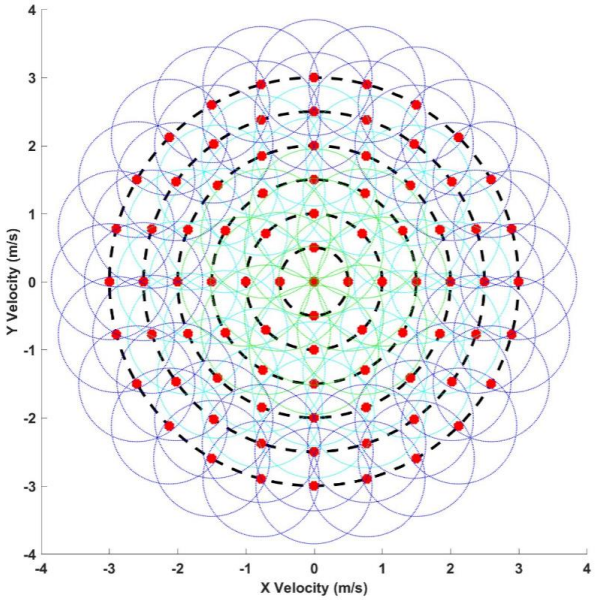


Figure 4.3. Depiction of the robustness of the controllers

A rough depiction of the robustness of the controllers is shown in Fig. 4.3. The controller in the interior (in green and cyan colors) are generally more robust than the ones on the periphery (in deep blue color) – which indicates that Q-Baller is running on high speed with more ferocious dynamics. However, all the controllers are capable of satisfying the gain scheduling requirement that for any pair of adjoint equilibrium points p_A and p_B and their stability domain D_A and D_B, there will be:

$$p_A \in D_A;\ \ p_B \in D_B \tag{4.28}$$

When the system is tracking a reference beyond the original starting point control area, the controller of the system will switch along its way to the reference. For example, when the controller attempts to reach velocity $\dot{X} = 2.121\ m/s$ and $\dot{Y} = 2.121\ m/s$, the system controller may switch through the path as shown in Fig. 4.4. There may be other possible paths, which is mainly depended on the designed switching judgement and the states of the system at that moment.

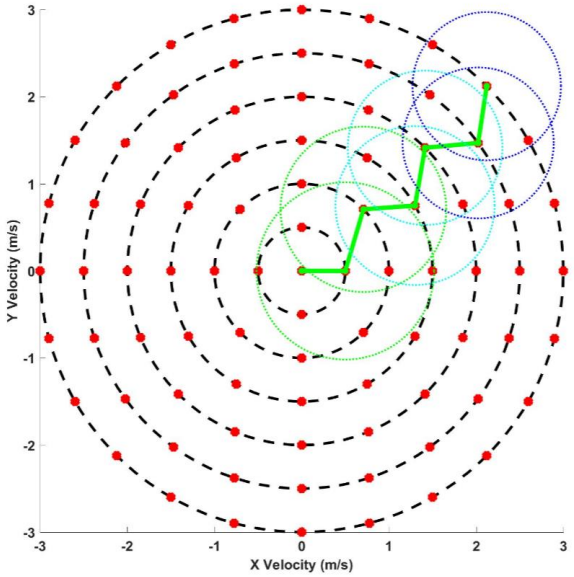


Figure 4.4. Depiction of the robustness of the controllers

4.4.3. Continuous Gain Scheduling Method based on N-Dimensional Delaunay Triangulation Weighting

The traditional gain-scheduled controller introduced in the previous subchapter is effective. However, it also has its deficiencies:

1) Like all gain-scheduled controller, the translation from an operating point to another is usually not smooth due to sudden changes of controller properties. Ramping of the property variation is usually applied to alleviate the problem. However, the algorithm for ramping is usually very complicated for multi-dimensional gain scheduling regarding different changing speed for different equilibrium states.

2) The controller performance may not be the best when the reference point is not the operating point, especially when the tracking point is in between the switching point between two operating points. The controller may have very poor performance and there may also be oscillation when the control algorithm is not designed perfectly.

The general idea of overcoming these problems is simple – make the variation of the controller continuous. But to realize this with the traditional gain-scheduled controller, it requires a huge amount of operating points in the control domain, which is usually very unpractical. Here, a new gain scheduling method is proposed to solve the problem with N-Dimensional Delaunay Triangulation Weighting Technique [4-3, 4-4].

Giving a set I_P of points in an N-Dimensional space, the N-Dimensional Delaunay Triangulation will create a Triangulation $Tri(I_P)$ which satisfies that:

1) No other point should be contained in the N-Dimensional circumsphere of any simplex in $Tri(I_P)$

2) No simplex in $Tri(I_P)$ is intersecting with other simplexes.

3) The combined volume of the simplexes is the convex hull of I_P

The idea of the Delaunay Triangulation is illustrated in Fig. 4.5, which presents the Delaunay Triangulations of random point sets in 2D and 3D respectively.

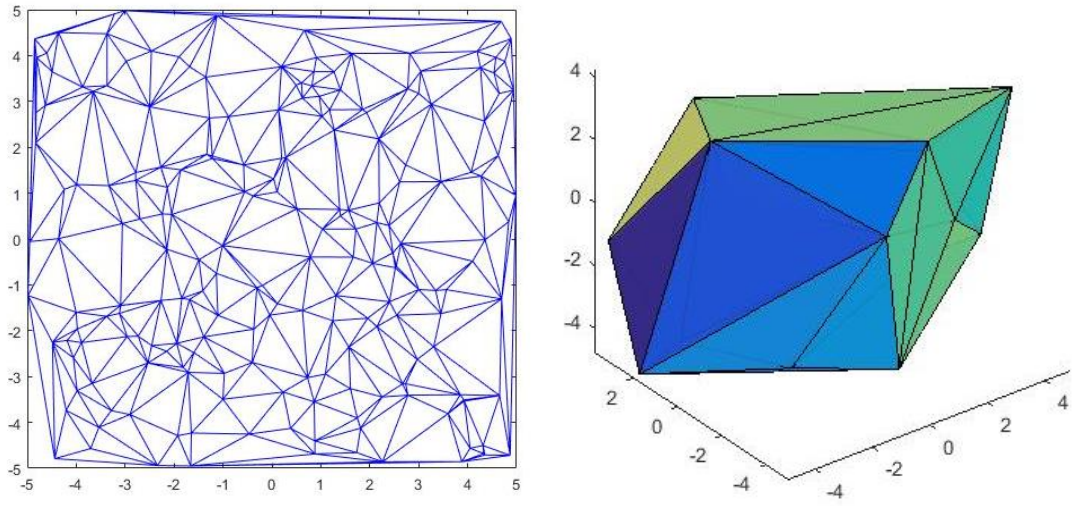


Figure 4.5: Random Point Delaunay Triangulation - Left: 2D; Right: 3D

The traditional Delaunay Triangulation application in Gain Scheduling requires the designer to carefully select operating points to prevent weak triangulations (when four of the points are the vertices of a rectangle). The previously selected operating points Q-Baller does not have such problem. The triangulation of the operating points for Q-Baller is shown in Fig. 4.6.

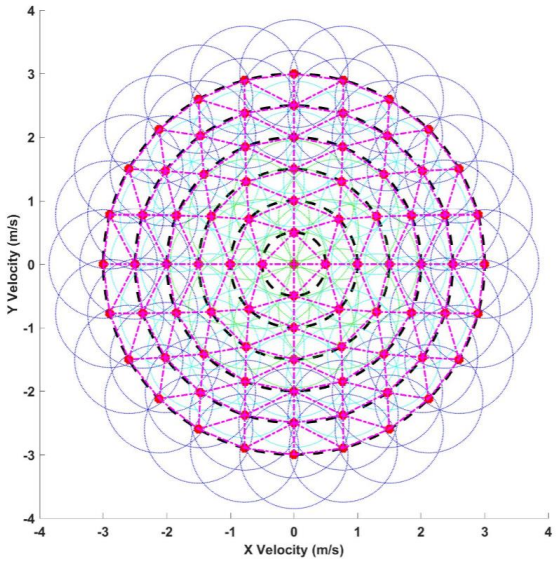


Figure 4.6. Delaunay Triangulation of the Operating Points

The weighting method based on triangulation uses the characteristic of the simplex, that is: for any N-Dimensional Simplex S_N whose vertices are $P_1, P_2, P_3 \dots P_{N+1}$, any extra single point P_X in the simplex will separate the N-Dimensional simplex volume V_S into N+1 parts named $V_{S_1}, V_{S_2} \dots V_{S_{N+1}}$. Each V_{S_n} is calculated when P_X replaces P_n. Then the Barycentric Coordinate [4-5, 4-6] of P_X in the simplex S_N is defined as:

$$Bary(P_X) = \left[\frac{V_{S_1}}{V_S} \quad \frac{V_{S_2}}{V_S} \quad \cdots \quad \frac{V_{S_3}}{V_S}\right] = [\delta_{B_1} \quad \delta_{B_2} \quad \cdots \quad \delta_{B_{N+1}}] \tag{4.32}$$

Obviously, the Barycentric Coordinate will satisfy:

$$sum\big(Bary(P_X)\big) = 1 \tag{4.33}$$

For any weighable information N_n that the vertices P_n hold, the information N_X can then be calculated as:

$$N_X = Bary(P_X) * \begin{bmatrix} N_1 \\ N_2 \\ \vdots \\ N_{N+1} \end{bmatrix} \tag{4.34}$$

To apply the weighting method to our controller, for the gain scheduled on a N-Dimensional space, the controller must satisfy the weighable condition which requires [4-6]:

$$A_{\mathcal{L}}(x_1, u_1) = A_{SS}(x_1, u_1) - B_{SS}(x_1, u_1)K(x_1, u_1)$$

$$\{P_S\} = \mathcal{L}\big(A_{\mathcal{L}}(x_1, u_1), x, Q_{\mathcal{L}_1}\big) \cap \mathcal{L}\big(A_{\mathcal{L}}(x_2, u_2), x, Q_{\mathcal{L}_2}\big) \cdots \mathcal{L}\big(A_{\mathcal{L}}(x_{N+1}, u_{N+1}), x, Q_{\mathcal{L}_{N+1}}\big) \neq \emptyset \tag{4.35}$$

Here $Q_{\mathcal{L}_n}$ is a random symmetric positive definite matrix and $x \in D_{CR}$ which is the control realm. The simplex encloses a control domain D_S must also satisfy:

$$D_S \subseteq D_{CR} \tag{4.36}$$

Then $\{P_{GS}\}$ can satisfy:

$$\{P_S\} \subseteq \mathcal{L}\left(\sum_{n=1}^{N+1} c_n A_{\mathcal{L}}(x_n, u_n), x, Q_{\mathcal{L}}\right) = \{P_{GS}\} \tag{4.37}$$

$$\{P_{GS}\} \subseteq \mathcal{L}\big(A_{\mathcal{L}}(x_1, u_1), x, Q_{\mathcal{L}_1}\big) \cup \mathcal{L}\big(A_{\mathcal{L}}(x_2, u_2), x, Q_{\mathcal{L}_2}\big) \cdots \mathcal{L}\big(A_{\mathcal{L}}(x_{N+1}, u_{N+1}), x, Q_{\mathcal{L}_{N+1}}\big) \tag{4.38}$$

Here, c_n are random positive real constants. Equation (4.35)~(4.38) can be explained as: if the controllers and operating points are well designed, the controller combined from the controllers designed at the operating points located at the vertices of a simplex can satisfy the stability requirement to control the state at any point enclosed in simplex. Therefore, the control input at x_{P_X} enclosed in S with a tracking reference x_r can be defined as:

$$u = -K_S(x_{P_X})(x_{P_X} - x_r) + u_S(x_{P_X}) = -\sum_{n=1}^{N+1} \delta_{B_n} K(x_n, u_n)\,(x_{P_X} - x_r) + \sum_{n=1}^{N+1} \delta_{B_n} u_e(x_n, u_n) \quad (4.39)$$

It should be point out that:

$$K_S \neq K(x_{P_X}, u_{P_X}) \text{ and } u_S(x_{P_X}) \neq u_{P_X} \quad (4.40)$$

Here u_{P_X} is the input that maintains the balance of the system at x_{P_X}, which is closely relative to the static state error of the controller. To solve this problem, for the velocity controllers of the gain scheduled states, integral controllers must be applied to eliminate the static state error.

For our Q-Baller, the gain scheduling is applied on a 2D space, where the 2-Simplexes are triangles. Controller of the Q-Baller at a point in the control domain is determined by the operating points at the 3 vertices of the smallest enclosing triangle of the point.

4.4.4. Operating Point Distribution base on Energy Level Sphere

The calculation of the enclosing simplex of the point usually requires a large amount of time if the gain scheduling points are not organized especially for high dimension gain scheduling. Therefore, a design method named as Energy Sphere Operating Point Distribution is proposed and applied to the controller.

The basic idea of the Energy Sphere Operating Point Distribution is to select the operating points on energy spheres defined as below:

$$x_{GS}^T P_{GS} x_{GS} = v_{GS}^2 \quad (4.41)$$

Here x_{GS} is a $M \times 1$ matrix containing the N gain scheduling states, P_{GS} is a $M \times M$ diagonal positive definite matrix, v_{GS} defined as the Energy Radius. The operating points of the system will be selected on the energy spheres of different Energy Radius, and later Delaunay Triangulation will be applied to operating points on the same spherical surfaces. For our Q-Baller, as introduced before, the energy radius is selected as the combined speed of the states, and operating points are selected on different speed circles.

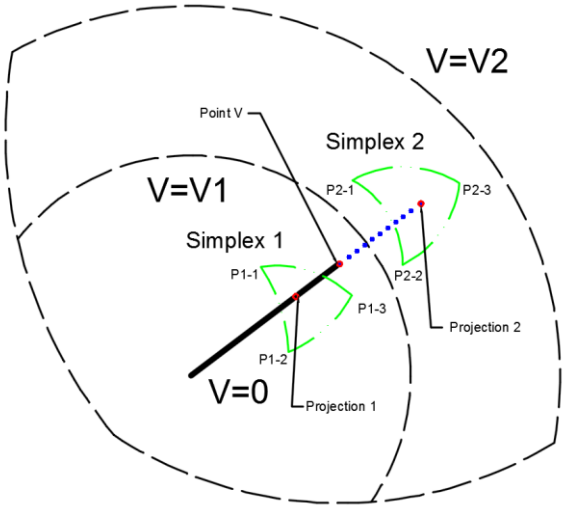


Figure 4.7. State Point projection on 3D Energy Spheres

For random points within the control domain, the points will be either on or in between energy spheres. The projections of the points along the radius onto the spheres will be enclosed by the smallest simplex from the operating points on the sphere surfaces. An example of this in a 3D gain scheduling is presented in Fig. 4.7.

To check if the projection of a point is on a certain N-Dimensional energy sphere simplex is easy, since the point that is enclosed by a sphere simplex will satisfy the following condition:

$$\begin{cases} c_n > 1 \ \ for \ \ n = 1,2,3 \dots (N+1) \\ c_1 + c_2 + c_3 + \cdots + c_{N+1} = 1 \\ \overrightarrow{OP_X} = c_1\overrightarrow{OP_1} + c_2\overrightarrow{OP_2} + c_3\overrightarrow{OP_3} + \cdots + c_{N+1}\overrightarrow{OP_{N+1}} \end{cases} \tag{4.41}$$

The operating points and the projection points are now on a sphere surface instead of a plane, and enclosing simplex is now a segment of the sphere surface. But we know that for any arc on a sphere, the arc, the center of the sphere and the chord of the arc are always on the same plane. The sphere segment can be projected on to the plane simplex made up by the chords, as illustrated by a 3D sphere example in Fig. 4.8.

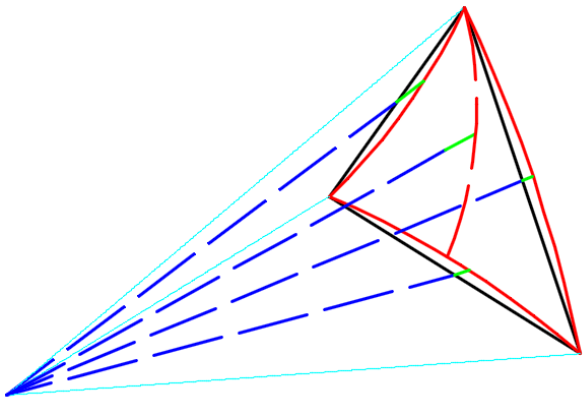


Figure 4.8. Projection of 3D Sphere Surface Simplex to Plane Simplex

The projection of a point on a N-Dimensional sphere simplex to the chord simplex along the radius is calculated by solving set of equation presented below:

$$\begin{cases} (1). \quad C = \theta_1 u_1 + \theta_2 u_2 + \cdots + \theta_N u_N \\ (2). \quad \begin{cases} C_1 = \partial_1 u_1 + u_2 \\ C_2 = \partial_2 u_1 + u_3 \\ C_3 = \partial_3 u_1 + u_4 \\ \vdots \\ C_{N-1} = \partial_{N-1} u_1 + u_N \end{cases} \end{cases} \quad => P_{Proj} = [u_1 \quad u_2 \quad \cdots \quad u_N] \tag{4.42}$$

Here equation set $(4.41).(1)$ is defined by the (N-1)-Dimensional chord simplex and $(4.41).(2)$ is determined by the radius that go through the point. Then the barycentric coordinates can be easily calculated with P_{Proj} and the vertices of the chord simplex.

However, it should be pointed out that if the operating points are scarcely distributed, the error between barycentric coordinates calculated from the chord simplex may be very different from the barycentric coordinates defined on a sphere surface. This problem can be overcome by slightly increase the operating points on the energy spheres.

For N-Dimensional Gain Scheduling, giving a random point P_V exists between two energy spheres E_{V_1} and E_{V_2}. The projection of P_V along the radius on the two spheres are P_{V_1} and P_{V_2}. The (N-1)-Simplexes S_{V_1} and S_{V_2} enclose the P_{V_1} and P_{V_2} respectively. The vertices of S_{V_n} are marked as $P_{V_{n,1}}$, $P_{V_{n,2}}$ ... $P_{V_{n,N}}$. Any weighable information N_V of P_V can therefore be calculated as:

$$W_1 = \left|\frac{V - V_2}{V_1 - V_2}\right| \text{ and } W_2 = \left|\frac{V - V_1}{V_1 - V_2}\right|$$

$$N_V = W_1 * Bary(S_{V_1}) * \begin{bmatrix} N_{V_{1,1}} \\ N_{V_{1,2}} \\ \vdots \\ N_{V_{1,N}} \end{bmatrix} + W_2 * Bary(S_{V_2}) * \begin{bmatrix} N_{V_{2,1}} \\ N_{V_{2,2}} \\ \vdots \\ N_{V_{2,N}} \end{bmatrix} \quad (4.43)$$

Similar to before, to perform the (4.41) for controller weighting, the two sets of operating points on the two sphere surface still have to satisfy (4.35) respectively. Moreover, this time the operating points also need to satisfy:

$$\{P_{S_V}\} = \{P_{S_{V_1}}\} \cap \{P_{S_{V_2}}\} \neq \emptyset \ when \ x \in D_{CR} \quad (4.44)$$

Providing that the control domain for the active pair of (N-1)-Simplexes is D_{VP}. This time, it is D_{VP} that must satisfy:

$$D_{VP} \subseteq D_{CR} \quad (4.45)$$

Then $\{P_{GS}\}$ can satisfy:

$$\{P_{S_V}\} \subseteq \mathcal{L}\left(\sum_{n=1}^{N} c_{1,n} A_{\mathcal{L}}(x_{V_{1,n}}, x_{V_{1,n}}) + c_{2,n} A_{\mathcal{L}}(x_{V_{2,n}}, x_{V_{2,n}}), x, Q_{\mathcal{L}}\right) = \{P_{GS}\} \quad (4.46)$$

If the controller and the operating points are well designed, the control input can be rewritten as:

$$u = -\sum_{m=1}^{2}\sum_{n=1}^{N} W_m \delta_{B_{m,n}} K(x_{m,n}, u_{m,n})\,(x_{P_X} - x_r) + \sum_{m=1}^{2}\sum_{n=1}^{N} W_m \delta_{B_{m,n}} u_e(x_{m,n}, u_{m,n}) \quad (4.47)$$

The advantages of gain scheduling through Energy Sphere Operating Point Distribution are listed below:

1) Since the system input is closely related to the energy of the system. The operating distribution through energy sphere is more reasonable. Different controller design parameter can then be easily applied to the operating points to acquire different performance at various energy level if necessary. Even for traditional triangulation weighting, distribution according to energy level will result to convenience in locating the simplex that contains the operating point.

2) The information of the controller at a random point in the control domain now comes as a combination of $2N$ adjoined controllers instead of $N + 1$ controllers, making the controller combination more informative especially for the points that would be located in the weak triangulated simplexes (for example when all the operating points are located on the intersections of rectangular grids).

3) Despite that the system now has to calculate 2 sets of Barycentric coordinates, the total calculation is still much smaller since the calculated Barycentric coordinates are now of (N-1) Dimension instead of N Dimensions. For a N-Dimensional simplex with vertices $P_0, P_1, P_2 \ldots P_N$, the volume of the simplex is calculated as:

$$\mathrm{V} = \left|\frac{1}{N!} det(P_1 - P_0, P_2 - P_0, \ldots, P_N - P_0)\right| \quad (4.48)$$

Here $det(P_1 - P_0, P_2 - P_0, \ldots, P_N - P_0)$ is a $n \times n$ determinant. Therefore, to calculate the Barycentric Coordinates of a N-Dimensional simplex costs $N^2/(N-1)$ times of that used to calculate the Barycentric Coordinates of a (N-1)-Dimensional simplex. Obviously, the time consummation gets significantly larger when N is larger.

The drawbacks of the application of Energy Sphere Operating Point Distribution are:

1) The requirement for controller design or the amount of the operating points are higher than the previous method.

2) The projection of operating points on the Chord Simplexes will lead to relatively large inaccuracy in the weighting of the controllers and equilibrium inputs when the operating points are scarcely located on a sphere

For our Q-Baller, when Energy Sphere Operating Point Distribution is implemented during gain scheduling. The energy spheres are 2-Dimensional, indicating that the 2D sphere simplexes are arcs on a circle, which makes the application of the previous methods extremely simple. The information of the controller at any point in the control domain is determined by at most 4 operating points.

4.4.5. Performance Comparison of the Gain-Scheduled Controllers

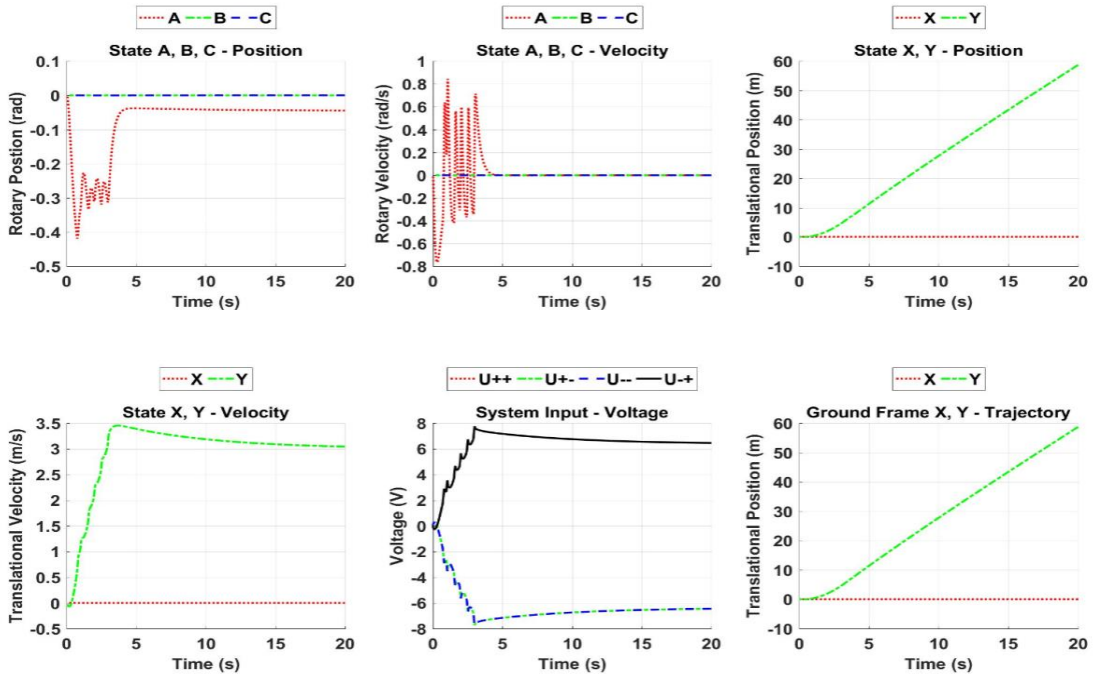
The 3 Gain-Scheduled controllers introduced before are compared through 2 experiments in this subchapter. The controllers are marked as GS Controller (Gain-Scheduled Controller), CGS Controller (Continuous Gain-Scheduled Controller) and ESD-CGS Controller (Energy-Spherically-Distributed Continuous Gain-Scheduled Controller) respectively for convenience.

As mentioned before, when controlled with the original Zero Point Linear Controllers, the system cannot stability beyond $1\ m/s$. The first experiment is a translational velocity control experiment. The system is controlled to from the Zero Point to accelerate to $3\ m/s$ at an acceleration of approximately $1\ m/s^2$ in Exp. 4.6.

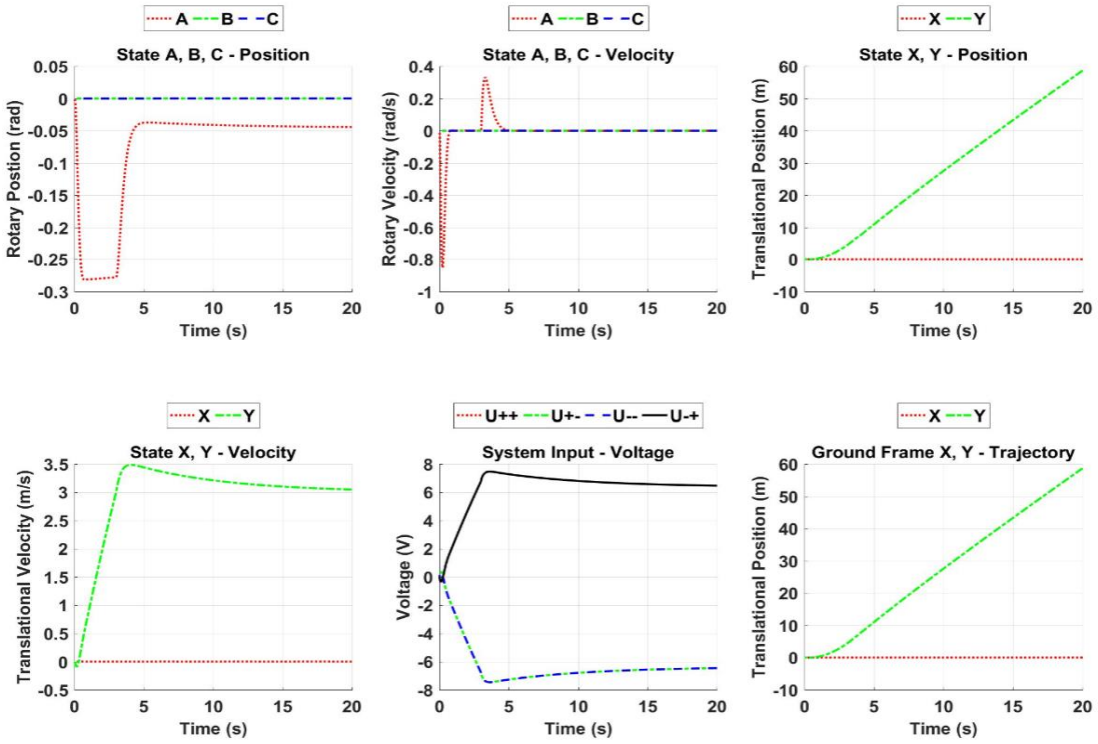
Experiment 4.6:

Tracking Reference: $A = 0;\ B = 0;\ \dot{C} = 0;\ \dot{X} = 3\ m/s;\ \dot{Y} = 0;$
Noise Simulation: Not Applied;
Observer Simulation: Not Applied;
Input Limitation: Power Rise time = 1 s; Maximum Voltage = 11.1 V;
Controller Specification: GS, CGS and ES-CGS Controllers; Ramped PD for A, B*; Ramped PI for* $\dot{C}, \dot{X}, \dot{Y};\ \ddot{Y} = 1\ m/s^2$.

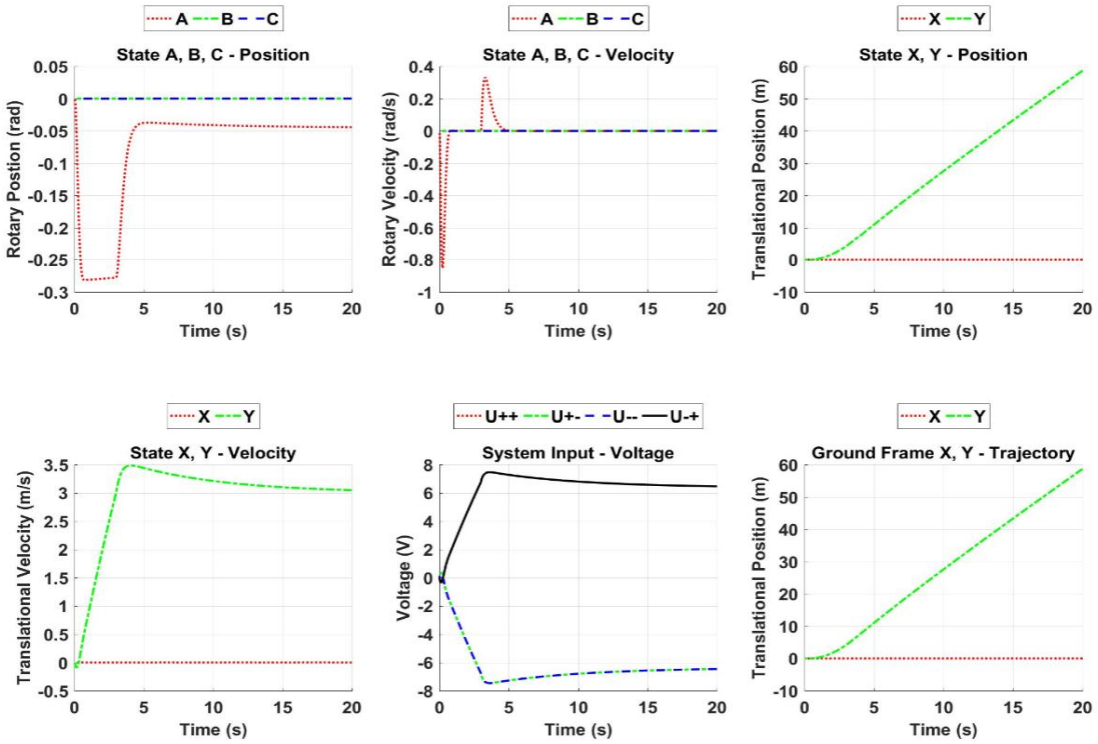
Result.1 (GS Controller):



Result.2 (CGS Controller):



Result.3 (ESD-CGS Controller):



The results from CGS Controller and ESD-CGS Controller are identical since the control algorithm effect are the same when the tracking reference is constantly on the connecting line of two operating points. We can also see the expected ripples from the outcome of the GS Controller Experiment as a result of switching between operating points. Exp. 4.6 proved that the CGS Controller and ESD-CGS Controller not only feasible but also plausible controller designs.

The second experiment only compared the performance of CGS Controller and ESD-CGS Controller of handling multiple velocity control in sequences. The system was commanded to:

1) Accelerate to $1.273\ m/s$ on X and Y directions respectively at a combined acceleration of $1\ m/s^2$; (From $t = 0\ to\ t = 1.8\ second$)
2) Rotate along C direction for 180 degrees in approximately 1 second. (From $t = 1.8\ to\ t = 2.8\ second$)
3) Accelerate to $2\ m/s$ on X direction and -1 m/s on Y direction at a combined acceleration of $1\ m/s^2$; (From $t = 5\ second\ to\ the\ end\ of\ simulation$)

To eliminate the overshoot, we have applied $P_{0.9}I$ Controller for the velocity controllers of $\dot{X}$ and $\dot{Y}$ as introduced before in Chapter 4.3.1. The result is presented in Exp. 4.7.

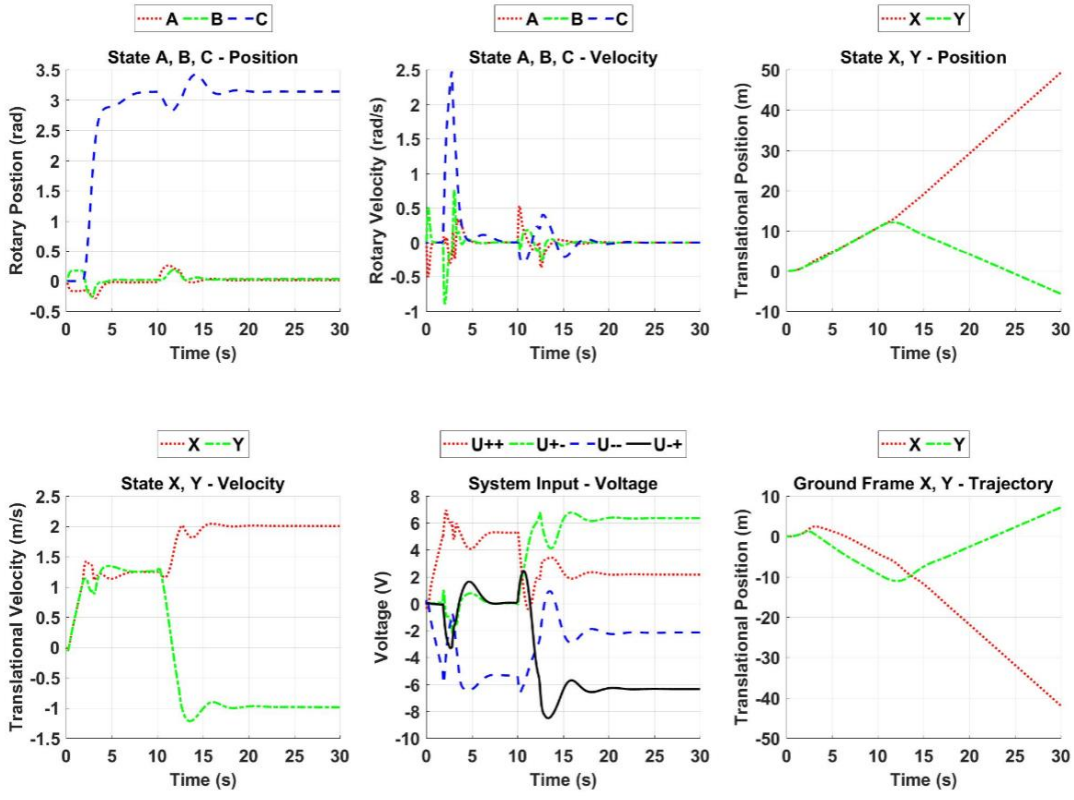
Experiment 4.7:

Tracking Reference:

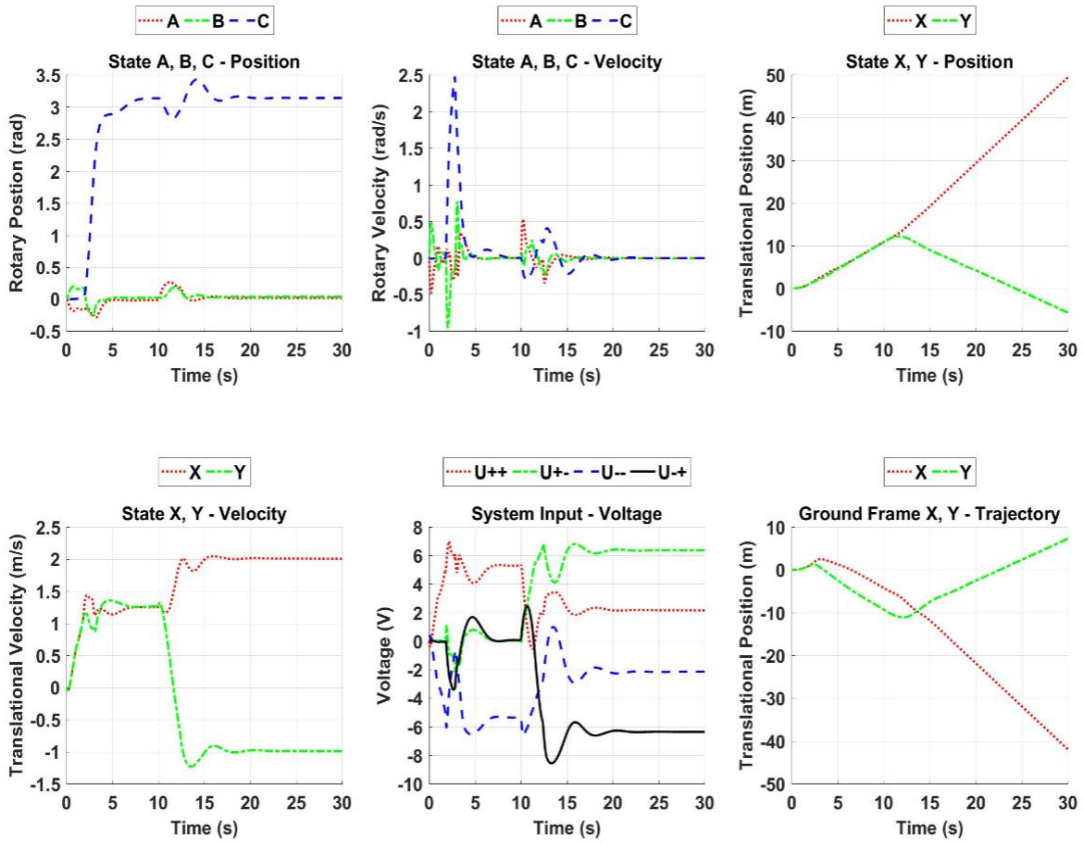
1) *Accelerate to* $1.273\ m/s$ *on X and Y directions respectively at a combined acceleration of* $1\ m/s^2$*; (From* $t = 0\ to\ t = 1.8\ second$*)*
2) *Rotate along C direction for 180 degrees in approximately 1 second. (From* $t = 1.8\ to\ t = 2.8\ second$*)*
3) *Accelerate to* $2\ m/s$ *on X direction and -1* m/s *on Y direction at a combined acceleration of* $1\ m/s^2$*; (From* $t = 5\ second\ to\ the\ end\ of\ simulation$*)*

Noise Simulation: Not Applied;
Observer Simulation: Not Applied;
Input Limitation: Power Rise time = 1 s; Maximum Voltage = 11.1 V;
Controller Specification: CGS and ES-CGS Controllers; Ramped PD for A, B*; Ramped* $P_{0.9}I$ *for* $\dot{C}, \dot{X}, \dot{Y}$*;*

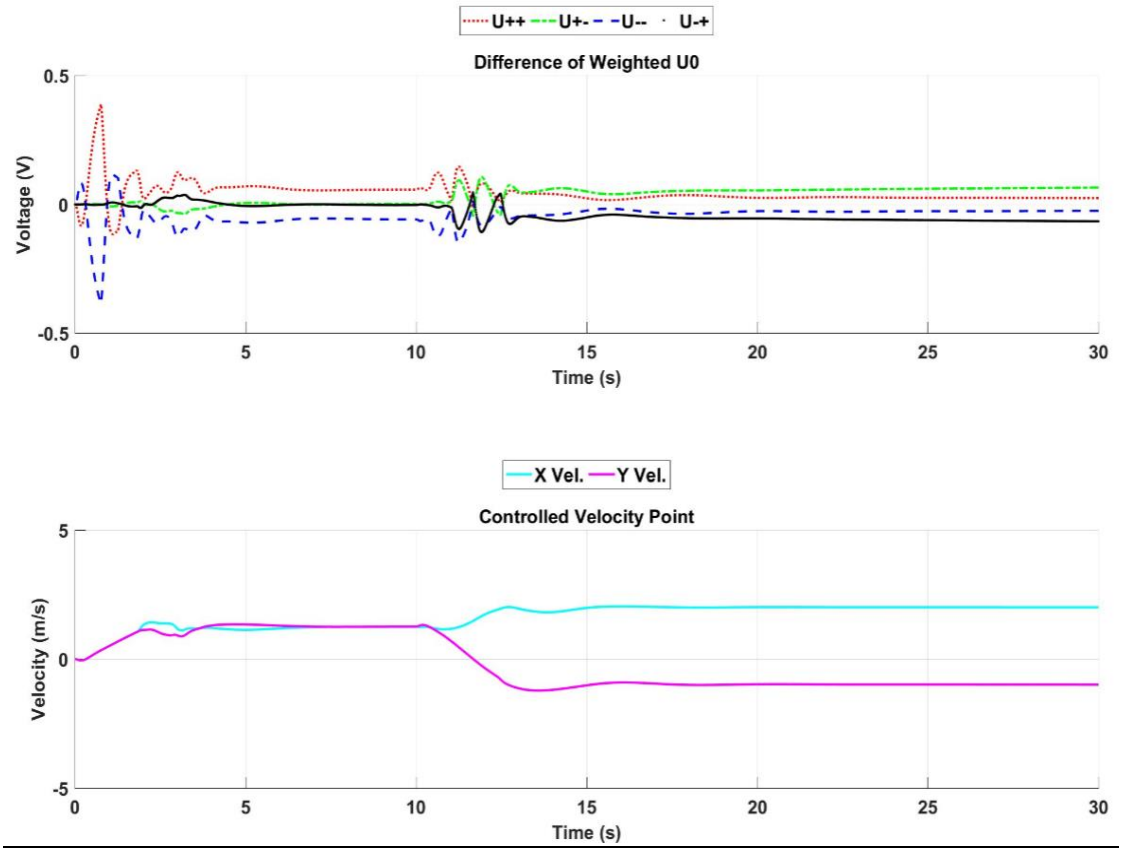
Result.1 (CGS Controller):



Result.2 (ESD-CGS Controller):



Result.3 (Difference Between Weighted Operating Point Inputs)



From the results it can be easily realized that the controller performances are very close. Both controller can keep the system stable and response fast in the control domain. However, it you pay close attention to the angular velocity performance you can see ripples in $\dot{A}$ and $\dot{B}$ the result of ESD-CGS experiment.

The ripples in the plot comes as a result of the inaccuracy of weighting for spherical projection described at the end of Chapter 4.4.4. The controller K_{GS} are not seriously affected, but the generated equilibrium inputs u_e are significantly different from the algorithm especially when the velocity is close to $v = 0.5\ m/s$ circle, where it only has 4 operating points on the energy sphere, as shown in Result. 3 of Exp. 4.7.

The ESD-CGS controller has a slightly weaker performance compared to CGS controller in Q-Baller's system control. Since gain scheduling for a system of high complexity and nonlinearity with a large domain of control usually will result in a large set of operating points, ESD-CGS should be more suitable for these systems since it has a much shorter calculation time. Regardless of the algorithm, the application of operating point distribution on energy level sphere has been proved plausible for the gain-scheduled control of Q-Baller.

4.5. Conclusion

Chapter 4 has discussed the stability and control of the system. A Linear LQR-PID Controller at the Zero Point which can keep the stability of the system around the Zero Point but cannot maintain balance beyond $1\ m/s$. A set of gain-scheduled controller is designed to cover a large domain of control.

In order to improve the performance and stability of the gain-scheduled controller, weighting method through Delaunay Triangulation has been applied to make the gain-scheduling continuous. The technique of operating point distribution on energy level sphere further improved the reasonableness of the operating point selection. Two weighting algorithms (CGS and ESD-CGS) were applied to the scheduled gains to test the performance, of which ESD-CGS is specially designed for the operating point distribution on energy level sphere technique. Both algorithms work well in maintaining the stability of the Q-Baller system.

The final controller design is able to reach $3\ m/s$ at an acceleration of $1\ m/s^2$. It is also able to keep a rotary velocity of $180\ degree/s$ when the combined translational velocity is below $2\ m/s$. Such control performance is satisfying enough for the flow up utility simulation.

Reference:

[4-1]. Hassan K. Khalil: *Nonlinear Systems (3rd Edition)*, ISBN-13:978-9332542037, Pearson Education (2002)

[4-2]. D. Subbaram Naidu: *Optimal Control Systems (1st Edition)*, ISBN-13: 978-0849308925, CRC Press (2012)

[4-3]. Boris Delaunay: *Sur la sphère vide*, Bulletin de l'Académie des Sciences de l'URSS, Classe des sciences mathématiques et naturelles. 6: 793–800 (1934)

[4-4]. Mark de Berg, Otfried Cheong; Marc van Kreveld; Mark Overmars: *Computational Geometry: Algorithms and Applications,* Springer-Verlag. ISBN 978-3-540-77973-5 (2008)

[4-5]. Einar Hille: *Analytic Function Theory Volume I*, 2nd Edition, 5th Printing, Chelsea Publishing Company, ISBN: 0-8284-0269-8 (1982)

[4-6]. Charles R. Johnson: *A Local Lyapunov Theorem and the Stability of Sums,* Linear Algebra and its Applications, Volume 13, Issure 1-2, P.37-43 (1976)

[4-7]. M. G. Kendall: *A Course in the Geometry of n Dimensions,* Dover Publications, ISBN-13: 978-0486439273 (2004)