

Motion Primitive with Aerial Manipulator for Dynamic Tennis Swing Motion

Fan Shi Moju Zhao

1. Introduction

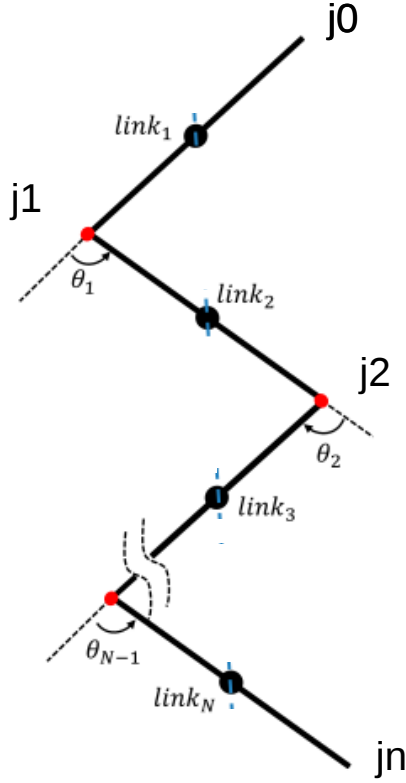


Fig.1 Hydrus structure

Base is the front end effector: j_0 .

Base's position and euler angle in world-fixed inertial reference frame:

$$p_b = [x_0, y_0, z_0], \phi = [\varphi_0, \theta_0, \psi_0] \quad (1)$$

State: $s = [x_0, y_0, z_0, \varphi_0, \theta_0, \psi_0, q_1, q_2, \dots, q_n]$.

$$p_{li} = p_b + R_b P_{bli}^b \quad (2)$$

$$\dot{p}_{li} = \dot{p}_b + R_b \dot{P}_{bli}^b - s(R_b P_{bli}^b) \omega \quad (3)$$

$$\omega = T_b \dot{\phi} = T_b [\dot{\varphi}_0, \dot{\theta}_0, \dot{\psi}_0] \quad (4)$$

$$\dot{p}_{bli}^b = J_p^{l_i} \dot{q} \quad (5)$$

$$\omega_{bli}^b = J_\omega^{l_i} \dot{q} \quad (6)$$

$$\omega_{li} = \omega_b + R_b J_\omega^{l_i} \dot{q} \quad (7)$$

$$K = \sum_{i=1}^n K_i \quad (8)$$

$$K_i = \frac{1}{2} m_{li} \dot{p}_{li}^T \dot{p}_{li} + \frac{1}{2} \omega_{li}^T R_b R_{li}^b H_{li} R_b^{l_i} R_b^T \omega_{li} \quad (9)$$

$$K = \frac{1}{2} \dot{s}^T D \dot{s} \quad (10)$$

$$D = \begin{bmatrix} D_{11}^{\{3*3\}} & D_{12}^{\{3*3\}} & D_{13}^{\{3*n\}} \\ D_{21}^{\{3*3\}} & D_{22}^{\{3*3\}} & D_{23}^{\{3*n\}} \\ D_{31}^{\{n*3\}} & D_{32}^{\{n*3\}} & D_{33}^{\{n*n\}} \end{bmatrix} \quad (11)$$

$$D_{11} = (\sum_{i=1}^n m_{li}) \quad (12)$$

$$D_{12} = D_{21}^T = - \sum_{i=1}^n (m_{li} S(R_b P_{bli}^b) T_b) \quad (13)$$

$$D_{13} = D_{31}^T = \sum_{i=1}^n (m_{li} R_b J_p^{l_i}) \quad (14)$$

$$D_{22} = \sum_{i=1}^n (m_{li} T_b^T s(R_b P_{bli}^b)^T s(R_b P_{bli}^b) T_b + T_b^T R_b R_{li}^b H_{li} R_b^{l_i} R_b^T T_b) \quad (15)$$

$$D_{23} = \sum_{i=1}^n (T_b^T R_b R_{li}^b H_{li} R_b^{l_i} J_\omega^{l_i} - m_{li} T_b^T s(R_b P_{bli}^b)^T R_b J_p^{l_i}) \quad (16)$$

$$D_{33} = \sum_{i=1}^n (m_{li} J_p^{l_i T} J_p^{l_i} + J_\omega^{l_i T} R_{li}^b H_{li} R_b^{l_i} J_\omega^{l_i}) \quad (17)$$

$$P_i = m_{li} g e_3^T p_{li} \quad (18)$$

$$P = \sum_{i=1}^n P_i \quad (19)$$

$$D(s)\ddot{s} + C(s, \dot{s})\dot{s} + g(s) = B(u)u \quad (20)$$

$$u = [f_1, f_2, \dots, f_n, q_1, q_2, \dots, q_n]^T \quad (21)$$

Here we use the quadrotor shape as the example:

$$B(u) = \begin{bmatrix} 0 & 0 & 0 & 0 & \mathbf{0}_{n \times 6} \\ 0 & 0 & 0 & 0 & \\ 1 & 1 & 1 & 1 & \\ 0 & d & 0 & -d & \\ -d & 0 & d & 0 & \\ c & -c & c & -c & \\ 0 & 0 & 0 & 0 & \mathbf{I}_{n \times n} \end{bmatrix} \quad (22)$$

$$\tau^{(root)} = \sum_{i=1}^4 (I_i^{(root)} \dot{\omega}_i^{(root)} + \omega_i^{(root)} \times I_i^{(root)} \omega_i^{(root)} + \dot{I}_i^{(root)} \omega_i^{(root)}) \quad (23)$$

$$\tau^{(root)} = \sum_{i=1}^4 l_{c_i}^{(root)} \times \left(\begin{bmatrix} 0 \\ 0 \\ u_i \end{bmatrix} - R_{world}^{root} \begin{bmatrix} 0 \\ 0 \\ m_i * g \end{bmatrix} \right) \quad (24)$$

$$\tau^{(root)} = \frac{\partial L^{(root)}}{\partial t} = \frac{\partial \sum_{i=1}^4 l_{c_i}^{(root)} \times (m_i v_i) + I_i \omega_i}{\partial t} \quad (25)$$

$$l_{c_i}^{(root)} = f_k(p^{(root)}, q) \quad (26)$$

$$v_i^{(root)} = v_{root}^{(root)} + J_p^i \dot{q} \quad (27)$$

$$\omega_i^{(root)} = \omega_{root}^{(root)} + J_q^i \dot{q} \quad (28)$$

$$\dot{\omega}_i^{(root)} = \dot{\omega}_{root}^{(root)} + J_q^i \ddot{q} \quad (29)$$

$$s = \begin{bmatrix} [p_x, p_y, p_z]^T \\ [v_x, v_y, v_z]^T \\ [\psi, \theta, \phi]^T \\ [w_x, w_y, w_z]^T \end{bmatrix}$$

$$\dot{s} = f(s, q, [u_1, u_2, u_3, u_4]^T, t) \quad (30)$$

$$\dot{s} = f(s, u) \quad (31)$$

$$s_{k+1} = s_k + f(s_k, u_k) \delta t \quad (32)$$

$$\delta s_{k+1} = A_k \delta s_k + B_k \delta u_k \quad (33)$$

$$\tau^{(cog)} = \sum_{i=1}^4 \frac{\partial I_i \omega_i^{(cog)}}{\partial t} = \sum_{i=1}^4 (I_i^{(cog)} \dot{\omega}_i^{(cog)} + \omega_i^{(cog)} \times I_i^{(cog)} \omega_i^{(cog)} + \dot{I}_i^{(cog)} \omega_i^{(cog)}) \quad (34)$$

$$\tau^{(cog)} = \sum_{i=1}^4 l_{c_i}^{(cog)} \times \left(\begin{bmatrix} 0 \\ 0 \\ u_i \end{bmatrix} - R_{world}^{cog} \begin{bmatrix} 0 \\ 0 \\ m_i * g \end{bmatrix} \right) \quad (35)$$

$$l_{c_i}^{(cog)} = f_k(p^{(cog)}, q) \quad (36)$$

$$v_i^{(cog)} = v_{cog}^{(cog)} + J_p^i \dot{q} \quad (37)$$

$$\omega_i^{(cog)} = \omega_{cog}^{(cog)} + J_q^i \dot{q} \quad (38)$$

$$a^{(world)} = \sum_{i=1}^4 R_{cog}^{world} \begin{bmatrix} 0 \\ 0 \\ \frac{u_i}{m} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (39)$$

Here, $I_i^{(root)}$ means the inertial tensor of i-th link, with respect to root link. We use parallel-axis method to calculate it, the distance from i-th link to root is the function of joints angles. So $I_i^{(root)}$ is time-variant.

$\omega_i^{(root)}$ means angular velocity of i-th link in root link frame.

$l_{c_i}^{(root)}$ means the distance from center of i-th link to the root.

2. Dynamic Tennis Swing Motion

2.1 Model

Geometric control researches showed the differential flatness of quadrotor system[1], which means that the output of 4 rotors rotation speed could be calculated based on the given input $[s_x, s_y, s_z, \quad]$, here s_x, s_y, s_z, s is the function $s_x(t), s_y(t), s_z(t), \quad (t)$ representing the position of robot in x, y, z axis and yaw angle at time t . The dynamics of the translational motion and rotation motion can be described as follows:

$$M^{\{W\}} \ddot{\mathbf{r}}_{CoG} = \begin{bmatrix} 0 \\ 0 \\ -M^{\{W\}} g \end{bmatrix} + {}^{\{W\}}R \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^N {}^{\{CoG\}}F_i \end{bmatrix} \quad (40)$$

$${}^{\{CoG\}}I_{multilink} \begin{bmatrix} {}^{\{CoG\}}\dot{w}_x \\ {}^{\{CoG\}}\dot{w}_y \\ {}^{\{CoG\}}\dot{w}_z \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^N {}^{\{CoG\}}y_i {}^{\{CoG\}}F_i \\ -\sum_{i=1}^N {}^{\{CoG\}}x_i {}^{\{CoG\}}F_i \\ \sum_{i=1}^N {}^{\{CoG\}}T_i \end{bmatrix} - \begin{bmatrix} {}^{\{CoG\}}w_x \\ {}^{\{CoG\}}w_y \\ {}^{\{CoG\}}w_z \end{bmatrix} \times {}^{\{CoG\}}I_{multilink} \begin{bmatrix} {}^{\{CoG\}}w_x \\ {}^{\{CoG\}}w_y \\ {}^{\{CoG\}}w_z \end{bmatrix} \quad (41)$$

2.2 Motion Primitive

In robotic trajectory planning researches, optimization and sampling based methods are two main-streams. Optimization based methods is usually time consuming, so in this paper we develop a rapid motion primitive algorithm based on sampling methods.

We develop a motion primitive method based on differential flatness quality, and 6-order polynomial functions are calculated in every axis and keep yaw angle to be fixed in the whole trajectory. Then jerk will be evaluated as the smoothness of the trajectory[1].

The motion primitive in each axis follows:

$$\mathbf{f}(t) = \begin{bmatrix} \mathbf{s}(t) \\ \mathbf{v}(t) \\ \mathbf{a}(t) \end{bmatrix} \quad (42)$$

$$\mathbf{f}(t) = \begin{bmatrix} \mathbf{a} & \mathbf{b} & \mathbf{c} & \mathbf{d} & \mathbf{e} & \mathbf{f} \\ 0 & 5\mathbf{a} & 4\mathbf{b} & \mathbf{c} & 2\mathbf{d} & \mathbf{e} \\ 0 & 0 & 20\mathbf{a} & 12\mathbf{b} & 6\mathbf{c} & 2\mathbf{d} \end{bmatrix} \begin{bmatrix} 1 \\ t \\ t^2 \\ t^3 \\ t^4 \\ t^5 \end{bmatrix} \quad (43)$$

For any random start point, we could acquire its position, velocity and acceleration from robot inertial data. The end point would be the hitting point, and its position, velocity and acceleration could be sampled in the work space. Considering a simple situation, we assume tennis ball is fixed, so only velocity and acceleration are needed to be sampled. For traverse time, we use traverse trajectory distance and average traverse speed to get the traverse time.

To get the faster swing speed, motion primitive for swing joint is also generated(Fig.2). The racket attached joint will backswing in advance, and then swing to guarantee racket will have velocity when being in collision with tennis ball. The backswing and swing time is decided by the selected trajectory of the robot.

2.3 Trajectory Selection

Considering the dynamic feasibility of the robot, the maximum velocity and acceleration have to be guaranteed. So the selector is as:

$$\min \int_0^T |s_x^{(3)}(t) + s_y^{(3)}(t) + s_z^{(3)}(t)|^2 dt \quad (44)$$

$$s.t. \quad V_{i_{min}} < s_i^{(1)}(t) < V_{i_{max}}, i = x, y, z$$

$$a_{i_{min}} < s_i^{(2)}(t) < a_{i_{max}}, i = x, y, z$$

$$0 < s_z(t)$$

Objective function is showed as Eq.(44), the calculus of the jerk 's absolute value from start time to end

time is calculated. The following constraints guarantee the dynamic constraints of robot and robot not crashing into the ground.

If there are other obstacles except for the ground in the environment, collision-free constraints should be considered as:

$$C_{obstacle} \cap (s_x(t), s_y(t), s_z(t)) = \emptyset \quad (45)$$

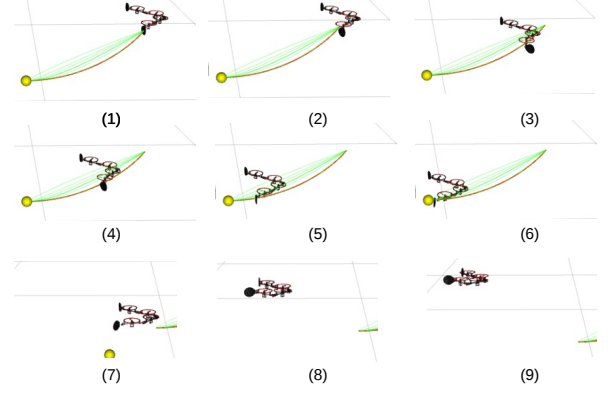


Fig.2 (1)-(6) is the swing motion based on motion primitive, green lines are primitive candidates, red line is the selected candidate, yellow ball is the hitting target. (1) is the start point, (2) begins to do backswing, (3) finishes backswing, (4) starts swing motion, (5) gets close to target and continue swing motion, (6) hits the target ball. (7)-(9) is the attitude control after hitting motion, robot recovers to be still and specific height.

3. Simulation Result

We implement this method into Gazebo simulation and nearly 5, 000 motion primitives could be generated in 100ms, and feasible trajectory could be easily found inside these candidates Fig.2. Then we build position and velocity control loop to guarantee the robot could follow the feasible trajectory.

4. Conclusion

This paper showed a rapid motion planning algorithm for Hydrus robot, and implement tennis swing task as the example to show the feasibility of the method. Experiment is implemented in simulation and the whole system is evaluated.

For future work, we are expecting to implement the method into real physical Hydrus robot to evaluate the whole system. And the use of GPU to accelerate motion primitive calculation is also considered. Perception of the tennis ball to generate catch-and-swing motion will be implemented as the future work.

Reference

- [1] Mark W Mueller, Markus Hehn, and Raffaello D'Andrea. A computationally efficient algorithm for state-to-state quadcopter trajectory generation and feasibility verification. In *Intelligent Robots and Systems (IROS), 2013*

IEEE/RSJ International Conference on, pp. 3480–3486.
IEEE, 2013.