LINEARIZATION-BASED MOTION CONTROL FOR MULTI-DRONE SLUNG LOAD SYSTEMS (MSLS)

Yongqing Liu[1], Zichen Zhang[1], Mohssen E. Elshaar[1], Alan Lynch[1], and Qing Zhao[1]

[1] *Electrical and Computer Engineering, University of Alberta, Edmonton, Canada.*

Email: alan.lynch@ualberta.ca

Introduction

We extend our previous work on MSLS motion control [1,2] in two ways: 1) the controller implementation was upgraded from ROS1 to ROS2, 2) a simpler quasi-static controller design is proposed, and 3) the control laws were validated in PX4 SITL environment.

Methods

Moving from ROS 1 to ROS 2 brings clear benefits: better real-time performance, more reliable middleware, and stronger long-term community support. With ROS 1, PX4 communication goes through MAVROS, which translates between ROS topics and MAVLink messages, adding an extra layer in the loop. In ROS 2, the PX4–ROS 2 bridge connects PX4's internal uORB messages directly to DDS/RTPS, cutting out that translation step. This reduces latency and makes the system more efficient. The full implementation accompanying this paper is openly available at: `github.com/yliu213/ACC25_ROS2`.

We propose a novel Quasi-Static Feedback (QSF) control and compare it to an existing linearization-based design using the Dynamic Extension Algorithm (DEA) [2]. Both controllers achieve linear exponentially stable error dynamics and the QSF has the advantage of a simpler controller expression. A subset of the geometric modelling notation is shown in Fig. 1. Further details are in [1,2]. The control objective is to track the output $y = (x_P^T, q_{13}, q_{21}, q_{22})^T$ where q_{ij} is the jth component of the pendulum attitude $q_i \in \mathbb{S}^2$. The system inputs are the forces generated by the drone $F_i \in \mathbb{R}^3$, which are transformed to a new input $u \in \mathbb{R}^6$. The result is a simplified model and tractable control design.

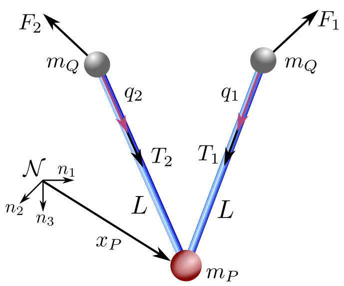

Figure 1: MSLS modelling.

The input–output relation is $\ddot{y} = Au + b$ with decoupling matrix A singular, and hence QSF achieves linearization with 3 subsystems:

$$[\ddot{y}_1, \ddot{y}_3]^T = A_1(q_1, q_2)\begin{bmatrix} u_{11} \\ u_{12} \end{bmatrix} + b_1,$$

$$[\ddot{y}_5, \ddot{y}_6]^T = A_2(q_2)\begin{bmatrix} u_{22} \\ u_{23} \end{bmatrix} + b_2,$$

$$[\ddot{y}_4,\, y_2^{(4)}]^T = A_3(q_1, q_2, \nu_2)\begin{bmatrix} u_{12} \\ u_{13} \end{bmatrix} + b_3,$$

Then, $u_i = A_i^{-1}(\nu_i - b_i)$, where $u_i, \nu_i, b_i \in \mathbb{R}^2$. **Key idea:** invert only $A_1, A_2, A_3 \in \mathbb{R}^{2\times 2}$, instead of the full 6×6. Auxiliary inputs $\nu_i = -K_i \begin{bmatrix} e_i \\ \vdots \\ e_i^{(j-1)} \end{bmatrix} + r_i^{(j)}$, where $e_i = y_i - r_i$ and r_i denotes the reference signal, with gains $K_i \in \mathbb{R}^{1\times j}$ chosen via LQR such that the linear error dynamics is exponentially stable.

Results and Discussion

Figure 2 shows payload tracking under QSF and DEA in PX4 SITL environment. Both reach the setpoints, with DEA achieving tighter tracking and QSF showing larger transients. In x_{p3} tracking, DEA exhibits an offset, while QSF regulates better.

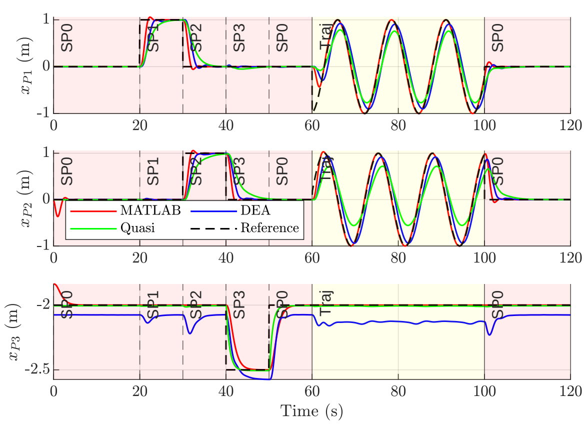

Figure 2: Payload position tracking (x_{P1}, x_{P2}, x_{P3}) under QSF and DEA controllers compared with the reference trajectory. Idealized MATLAB simulations are shown for comparison.

Conclusion

Two exact-linearization controllers were developed for the MSLS: QSF and DEA. QSF offers a simple algebraic design suited for implementation, achieving better overall performance.

References

[1] PX4 Dev Team. ROS 2 User Guide.
[2] Jiang Z, et al. J Intell Robot Syst 109(2): 42, 2023.
[3] Al Lawati M, et al. Am Control Conf: 1–7, 2025.