# A Survey of Quadrotor UAVs, Gupte et al. (2012)

* Introduction: different types of craft, usage
* Mechanism of flying: airframe design, motion about roll/pitch/yaw controlled via throttle
* Control systems and dynamic models
  + E.g. PID, backstepping, neural networks, feedback linearization
  + Sensors: IMU, GPS, cameras
  + Multi-agent environments, cooperation
  + Nature and interactions with it
* Sensors:
  + IMU = accelerometer + gyroscope
  + Ultrasonic range sensors for low-level altitude control, obstacle avoidance
  + IR, laser range finders
  + Barometer: altitude
  + Magnetometer: direction
  + GPS: tracking and localization
  + Camera: image recognition, obstacle avoidance
* Vision:
  + Object detection, tracking, pose estimation, navigation, obstacle detection etc.
  + Visual servoing: feedback to control robot motion
  + Offboard mocap with blob detection
  + Sensor fusion with IMU
* Navigation: SLAM, algorithms must be offboard and 3D
* Applications: search and rescue, surveillance, military purposes

## Read next

Feasibility considerations in formal control … Maithripala et al 2008

**A formal model approach for the analysis … Tsourdos et al, 2005**

Formation control numerical simulations … Xue et al, 2009

Feedback Linearization and High order … Benallegue et al, 2008

PID vs LQ control techniques applied to … Bouabdallah et al, 2004

Backstepping and sliding-mode techniques … Bouabdallah et al, 2005

Modeling and PD control of a quadrotor … Erginer et al, 2007

Multiagent quadrotor testbed … Waslander et al, 2005

# Introduction to the Special Issue on Aerial Manipulation, Ruggiero et al. (2018)

* Introduction:
  + Applications
  + Challenges (close proximity, underactuation of UAV, unstable dynamics, aerodynamic effects, dynamic effects due to object)
  + Outstanding issues: energy, safety, lack of high accuracy, adaptability to real world
* Outline
  + Ruggiero et al. survey the field
  + Single arm aerial manipulation:
    - Tognon et al. : control-aware planner – controller (2nd order inverse kinematics algorithm + dynamic tracker) used as local planner, predicts behavior of controller, guarantees correct execution of maneuvers
  + Cooperative aerial manipulation:
    - Loianno et al: cooperative transportation of objects to an end zone
    - Six et al: cooperative aerial manipulation with rigid platform, dynamic model, cascaded controller
    - Lee et al: online estimation of unknown payload’s mass and inertia, adaptive controller, obstacle avoidance
    - Tognon et al: CPT, cable-suspended load, no explicit communication, admittance controller, uses cable forces, positive internal force to control both position and orientation

## Read Next

Aerial Manipulation: a literature review, Ruggiero et al, 2018

Control-aware motion planning for …, Tognon et al, 2018

Aerial co-manipulation with cables …, Tognon et al, 2018

The kinematics, dynamics …, Six et al, 2018

An integrated framework for cooperative aerial manipulators …, Lee et al, 2018

# Aerial Manipulation – a literature survey, Khamseh et al. (2018)

* Introduction
  + Application of UAVs
  + Aerial manipulation, contrast with ground-based manipulation
  + Survey of control algorithms for UAVs, motion- and trajectory-planning for UAVs, path planning in the presence of disturbances, past efforts in UAM
* Physical subsystems:
  + Past technology,
  + Rotary-wing UAMS, airships (issues)
  + Quadrotors (benefits)
  + UAV platform characteristics (low payload weight budget and consequence for attaching manipulators)
  + Interaction mechanisms
    - Robotic manipulator: most common, UAV gets in way of manipulator
    - Gripper attached to UAV body: previously common
    - UAV body / UAV with rigid tool: efficient for large force exertion
    - Tether: tensile forces only, common for load transportation, ineffective for force exertion or pick and place
  + Sensory configurations
    - IMUs = 3-axis accelerometer, 3-axis gyroscope, 3-axis magnetometer; feeds into UKF / EKF
    - GPS, altimeters, ultrasonic distance sensors
    - Indoors, visual methods more efficient than GPS
* Missions and operational scenarios
  + Missions
    - Aerial manipulation:
      * Load transportation
      * Methods:
        + Tether: derived from slung-load systems research, alleviates need for extra mechanism and increases allowable payload weight, prevents pickup process automation and stability of motion may be compromised by load swings
        + Load picked up by gripper or manipulator
    - Force/torque exertion on the environment, applications
    - Assembly and structural construction, applications
    - Other missions
  + AM Scenarios:
    - UAM flies to object, locks manipulation mechanism when near object (simplifies motion control problem but extends operation time), manipulator is actuated to grasp object
    - UAV and manipulator operate simultaneously – reduces time needed for task, so is more advantageous than the first technique
* Modeling a UAM
  + Definitions of inertial coordinate frame, vehicle CoM’s coordinate frame, equation of motion for end effector
  + Motion equations derived using recursive Newton-Euler (more convenient for actual implementation) or Euler-Lagrange (better for analysis of dynamics and control design)
  + Description of Recursive Newton-Euler method: quadrotor and manipulator are two different subsystems interacting. Model accounts for the static and dynamic effects of the manipulator, and for the fact that the thrust force in the UAM generates undesirable torques as the system’s CoM moves away from the quadrotor CoM.
  + Euler-Lagrange method: quadrotor + manipulator are one system; equations of motion; system dynamics equation; UAM model elements are configuration-dependent and time-varying; imperfect knowledge of mass and inertia matrices, aerodynamic models introduce many uncertainties
  + Interactions with environment
* Estimation and control of UAM:
  + State and parameter estimation
    - Least square problem using control input and acceleration measurements to estimate mass, CoM, inertia of a UAM-grasped load // to estimate UAM CoM: adequate for static scenarios but dynamic changes from arm movement degrade performance
    - Carrying unknown mass (doesn’t need to hang beneath)
    - EKF most widely-used state estimation algorithm for UAVs, performance is degraded for very non-linear systems
    - UKF: better performance, higher computational complexity; spherical UKF has lower complexity and lower performance
    - Visual state estimation, fusion with GPS measurements, optical flow
  + Position control:
    - Two paradigms: decoupled or coupled
    - Control algorithms for decoupled: PD/PID (simple, with gain scheduling), feedback linearization, backstepping, adaptive control
    - Control algorithms for coupled: feedback linearization, impedance control, non-linear backstepping, LQR, adaptive control, MPC
    - Decoupled control design:
      * Effects of manipulator on UAV treated as disturbances, not explicitly considered in controller design
      * Satisfactory behavior in simple scenarios, performance degraded in more realistic applications
      * Variable parameter integral backstepping: note the paradigm of computing quadrotor MoI and CoM as functions of manipulator joint angles + compensating for quadrotor motion in manipulator’s controller
      * Adaptive control useful for systematic treatment of time-varying parameters (Lyapunov-based method, PID + adaptive control, PID + Model reference adaptive control + 2 gain-scheduling schemes)
      * Non adaptive PID schemes can lead to UAM instability
    - Coupled control design:
      * Equations of motion of combined dynamical model already account for nonlinear and coupled dynamics, unified model-based controller design difficult and depends on computational resources
  + Force and impedance control (for manipulation)
  + Vision-based control: state estimation, object tracking, applications
  + UAM control in presence of gusts:
    - Good control in low to medium-speed gusts
    - Wind gusts degrade control quality
    - Position errors must be addressed for slung-load systems in gusts
  + Teleoperation: human operators can’t manually control UAMs well, so computer-aided control is necessary; humans can share in tasks with UAM control systems
  + Human-UAM interaction
* Discussion and future directions
  + Variety of platforms; rotary wing vehicles best suited to AM
  + Limitations on UAV payload budget and computational resources
  + Physical interaction: tether (limited to load transportation), actuated robotic manipulators
  + Missions: load transportation, inspection, pick and place, etc.
  + Modeling dynamics: recursive Newton-Euler (actual implementation), Euler-Lagrange (analysis of dynamics)
  + Completely manual control impractical, hybrid manual and autonomous control of UAMs yet to take off
  + Challenges obstruct teleoperation of UAMS (imperfect communication, time-varying delay, information loss)
  + Vision used for state and pose estimation, target tracking
  + Visual servoing of UAMs is developing
  + Decoupled control is simpler theoretically, more time-consuming, less accurate control
  + Coupled control is more complex, more efficient, more accurate control

## Read Next

* Control algorithms: 16
  + LQR: 63
  + Adaptive Control: 15, 47, 95, 146
  + MPC: 24, 86
  + PID: 15, 64, 87
  + Feedback linearization: 24, 75
  + Backstepping: 42, 43
  + Motion/trajectory planning: 17, 18
* Load transportation: 10, 11, 22, 49, 97
  + With gusts: 103, 159
* Modeling: 11, 15, 23
  + With unknown parameters: 135

# An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation, Kelly (2017)

What is trajectory?

Trajectory optimization. Key definitions for T.O. problem. Direct collocation: discretize the T.O.P., approximate using polynomial splines, convert into non-linear program. Non-linear programming.

Example: Block move with minimum force. Problem statement: system dynamics, boundary constraints, objective function (integral of control effort squared). Analytic solution: modify cost functional to include Lagrange multiplier; T.O. requires first variation = 0, second variation >=0; derivation of conditions for optimality with respect to Hamiltonian expression, find Lagrangian, solve for control functions, solve for system dynamics. Trapezoidal collocation: discretize, convert system dynamics into constraints using trapezoidal quadrature, approximate objective function using T.Q. Initialize NL programming solver with guess for x, v, u. Interpolation with splines for control and state.

Trapezoidal collocation. T.Q. for integrals. System dynamics using T.Q. Constraints enforced at the collocation points. Interpolation with linear splines: integrate for state and solve using boundary conditions on state.

Hermite-Simpson collocation: S.Q. for integrals. System dynamics using S.Q., must use a second collocation equation for midpoint values, problem formulation can be compressed (better for many segments) or separated (better for fewer segments). Constraints enforced at all collocation points (including midpoints). Interpolation with quadratic splines.

Practical considerations. Initialization: effect of constraints, problem-specific knowledge is valuable, simply assume trajectory is a straight line in state space, simplify T.O.P until a simple technique yields reasonable answer then use this solution to initialize original problem. Mesh refinement: denser time mesh and higher-order method increase accuracy and computation, start coarse and get finer, check error over segment then subdivide or increase polynomial order in segment (‘hp-adaptive meshing’). Error analysis: transcription errors arise due to discretization choice (in method or grid), check error by subtracting system dynamics along candidate trajectory from state derivative. Code debugging: family of optimal solutions (introduce regularization term to cost function), non-smooth solution so slow NL program convergence (mesh refinement, smoothen problem, use multiphase method), inconsistent objective and constraint functions, infeasible problem (either problem impossible or initialized with bad guess), solution reached (for global optimality check with many initial guesses and different transcription methods that this solution is reached). Function consistency.

Example: Cart-Pole Swing-up. System dynamics, state space representation. Objective function: integral of actuator effort squared, creates smooth trajectories which are faster and accurate to solve with polynomial transcription methods, easier to stabilize with conventional controllers. Boundary constraints. State and control bounds. Attempted solution using T.C., H.S.C. Initialization: naïve approach (linear motion from initial to final state), uniform grid. Results: compute new grid with shorter segments near discontinuities.

Example: 5-Link Bipedal Walking. Model. System dynamics, state space representation. Objective function: torque-squared cost function. Constraints: don’t introduce redundant constraints. Initialization. Results: run optimization with coarse mesh, then fine mesh.

Example: Minimum-work block move. Reconsider block move problem, introduce new objective function. Problem statement. Analytic solution using bang-bang control law. Discontinuities: use slack variables or smoothen problem. Initialization: naïve approach. Slack variables: solution is still discontinuous, extra slack variables increases size of NL program, causes slower convergence, method and initialization. Smoothing: smaller parameters approximate original function better, so increased accuracy and solution difficulty, conduct convergence tests with smaller parameters to guarantee solution correctness. Results: discontinuity causes slow convergence, slack variables with fine mesh take long time, increasing smoothing decreases computation time and accuracy.

Overview of related topics and software packages. Trajectory optimization v parameter optimization. Open loop v Closed loop solutions: T.O looks for O.L. solution to optimal control problem, best for large-dimensional systems, large state space and high accuracy. Dynamic programming looks for an optimal policy (C.L. solution), best for lower-dimensional systems, computing scales exponentially with dimension. Continuous time v Discrete time. Indirect Methods: analytically find necessary and sufficient conditions for optimality then discretize conditions and solve numerically, generally more accurate, region of convergence smaller so needs better initialization, difficult to construct conditions analytically. Shooting methods: direct single shooting; direct multiple shooting; path constraints difficult to implement, high nonlinearity so potentially poor convergence. Orthogonal collocation, trajectory calculated using barycentric interpolation. Differential dynamic programming. Multiphase methods: easier to compute and more accurate for hybrid system trajectories. Thru-contact methods for systems with contact mechanics and unknown sequence of continuous motion phases. Merits of each method: compute solution approximation with a direct method, use it as an initialization for an indirect method; start with low-order collocation method for initial solution, perform error analysis, change mesh size and order accordingly.

Electronic supplement (MATLAB Library and worked problems).

Analytic solution to block move example in section 2.

Derivation of Simpson Quadrature.

Orthogonal polynomials overview.

Parameters for example problems.

Biped dynamics.

# A Survey on Aerial Swarm Robotics, Chung et al. (2018)

Introduction: swarms, challenges of UAVs, 3D flow and swarm autonomy, hierarchy architecture.

Models: stability and control of swarms, types of multiagent systems (team, formation, swarm), models for swarm dynamical systems, physics-based models for robotic agents, synchronization with leader following, leader selection and sensor placement (can use submodular optimization to choose a leader to maximize and objective function), synchronization and hierarchical stability for swarms (analysis considering graph theory and stability).

Control of swarms in 3D worlds: two task classes (exploring environment, traversing environment to complete a goal), objectives and practical considerations;

trajectory generation and motion planning: with or without task allocation, possibility of real-time solutions for constrained optimal control problems, format of the optimal control problem and similarity to MPC, objectives for multirobot path planning, NP-hard for exact optimal solution v. suboptimal solutions found with heuristics, direct optimal control v. preselected geometric path, design of control laws (find control input values via trajectory generation with MPC; separate the optimized state trajectory, as desired trajectory for a tracking controller), coordination and collision-avoidance require communication and/or sensing other agents, using a hierarchical model to deal with exigencies, dealing with actuator failures and battery depletion.

Simultaneous planning with distributed assignment (swarm considerations);

Collision avoidance and collision free motion: intuition (speed adjustment, sequential trajectory re-planning), artificial potential fields, barrier functions, mixed integer linear program.

Aerial manipulation (cable suspension, rigid grasp), CPT (with explicit or implicit communication), minimize internal force to conserve energy.

External control of aerial swarms (real-time assignment of objectives by external user, interaction with a user-controlled adversary), control of a virtual leader, virtual rigid body framework, using dynamic set of leaders v. fixed leader, human interfaces, adversarial control (herding, containment), working with unknown flock models.

Aerial distributed sensing, monitoring and cooperative mapping: focus on target with target search and tracking, focus on area with surveillance and modelling, cooperative aerial mapping and SLAM.

Technology for swarming: platforms, vehicle power management, pose and state estimation (external sensors, onboard sensors), communication infrastructure (centralized or p2p).

Conclusions and future work: decision-making architectures, adversarial control, interdependency between **swarm vehicle dynamics, uncertainties and swarm learning/control methods.**

## Read next:

Trajectory generation and motion planning

Direct optimal control with MPS: 25, 85-87

Sequential convex programming: 25, 87

Observable Markov decision process: 88, 89

Objectives: 90

Separate optimized trajectory from tracker: 25, 28, 87, 93, 94

Collision avoidance:

Mixed integer linear programming: 85, 86, 119

Potential fields: 121-124

Barrier functions: 125, 126

Aerial manipulation:

Cable-suspension: 129-133

Rigid grasp: 134-138

CPT: 129-132, 135-137. 132 for consideration of internal forces, 136 minimizes internal forces

“Adaptive control of a quadrotor UAV transporting a cable-suspended load of unknown mass”, Dai et al. (2014)

# Load Transportation Using Quadrotors: a survey of experimental results, Villa et al. (2018)

Introduction: applications, challenges, cable suspension v. grasping.

Grasping v. cable suspension: CS – increase DoF of UAV, maths model is difficult, increased agility, decreased stabilizability, switches between cable modes. Grasping: decreased agility, increased energy consumption, increased mass and rotational inertia.

CS Loads:

Swing-free

* 5: dynamic programming on a discrete linearization of the system model using an adaptive controller with feedback linearization; lag-lead control with PID control for heading angle.
* 6: reinforcement learning, Markov decision process and greedy algorithm for swing-free optimal trajectory.

Hybrid model

* 7: suspended load allowed to have large swings under active feedback control, finite time durations with zero cable tension
* 9: setup, pull and raise modes

Aggressive maneuvers with hybrid model: 11 considers quadrotor with load (taut cable) and quadrotor alone (load in freefall), 2D trajectory planning with obstacle avoidance using MIQP.

Aggressive maneuvers without hybrid model: 12 treats trajectory planning problem as Mathematical Programming with Complementary constraints, solves with SQP. Outperforms state of the art with respect to speed without restricting trajectory complexity. Simple parametrization for different tasks. Guarantees trajectory feasibility with respect to system dynamics and control input bounds.

Aggressive maneuvers for CPT

* 18: trajectory generation for multiagent team, feasible trajectory for obstacle-free 3D workspaces, motion planning using iterative geometric motion planning based on HOOP (19)
* 20: geometric controller

Vision-based load transportation: 3 – precise and fast payload positioning with minimum swing using Interconnection and Damping Assignment Passivity-Based Control.

21: leader-follower setup, decoupled QUAV with load, quadrotor and load control by LQR, load tracking via PD controller, trajectory planning using straight-line trajectory at constant height.

Grasped load:

* Vision-based CPT: 13, 14, 17, 22
* UAV-ground vehicle CPT: 23, 24
* Cooperative manipulation: 25