Autonomous Battery Replacement for Quadcopters

Hariharan Ravichandran
MS Robotic Systems Development
Carnegie Mellon University
Pittsburgh, USA
hravicha@andrew.cmu.edu

Anishram Senathi
MS Robotic Systems Development
Carnegie Mellon University
Pittsburgh, USA
asenathi@andrew.cmu.edu

Dhruv Tyagi
MS Robotic Systems Development
Carnegie Mellon University
Pittsburgh, USA
dmtyagi@andrew.cmu.edu

***Abstract*—With delivery drones being deployed in multiple cities, one key aspect of the logistics is constantly replacing the drone batteries. This report details an autonomous robotic battery replacement solution for the delivery drones, which can be employed to work constantly throughout the day, ensuring high operational efficiency of the overarching delivery infrastructure.**

***Index Terms*—autonomous, battery, replacement, quadcopters, robot, arm**

I. Introduction

E-commerce corporations and the logistics industry around the globe are racing to adopt widespread usage of drone delivery solutions for the last mile delivery segment of the supply chain [1], [2]. Quadcopters and drones are not subject to the same constraints as the currently employed ground motor vehicles are. They can transport light-weight and small sized packages over intracity distances much faster and cheaper than wheeled vehicles. Although the drone solutions are extremely efficient for this goal, they have a separate constraint imposed upon them due to the limitations of present battery technology. An average quadcopter possesses a battery runtime of approximately 30 minutes. To continue to deliver packages to city-wide locations, the quadcopters would need to recharge or replace their power source numerous times throughout the day. This poses an obstacle to efficient logistical operations, especially if manual intervention is required to replace discharged batteries with the freshly charged ones.

The procedure of replacing quadcopter batteries is repetitive and can be streamlined using autonomous solutions [3]. In this report, one such implementation is discussed in detail. The solution employs a robotic arm that uses a camera to detect the battery to be replaced, manipulates the battery to store it and then inserts a new battery into the battery slot.

II. Key challenges

Utilizing an autonomous robotic solution for the application of battery replacement presents challenges that are required to be addressed to ensure that the solution is reliable, robust and is accommodating of varying scenarios.

A. Low tolerance

The battery has to be fit securely in the compartment which holds it. This ensures that the electrical connections that supply the battery's power to the rest of the system always remain in contact with the battery terminals while also ensuring that there is no movement of the battery mass that disturbs the quadcopter's operation. Fitting the battery securely also avoids any risk of the battery destabilizing due to aggressive movements. To fit the battery securely, the battery compartment or slot is required to be the exact size and shape that can accommodate the whole battery while also minimizing gaps. Any robotic solution will have to successfully extract and insert the battery while obeying the low tolerances of the battery slot. This demands high accuracy, especially from the insertion maneuver of the system.

B. Smooth Operation

Batteries used to operate quadcopters are constructed to be dense with electrical power while also possessing low weight. Commonly used drone batteries contain elements which may lose efficiency or functionality if not handled with attention. This requires any battery replacement solution to have safe and smooth operation to minimize the likelihood of dropping or damaging the battery.

C. Quadcopter Landing Positions

Due to various factors like wind conditions, sensor feedback accuracy, and speed, aerial vehicles have a lower accuracy of position and orientation while landing. This causes a lot of variability in the system, there are numerous positions and poses in which the drone can land in the designated landing zone. For a robotic solution to be reliable, it needs to accommodate for these varying possibilities while replacing the batteries.

D. Compliance with Locking Mechanism

As mentioned in subsection II-A, the battery is required to be secured in place while powering the quadcopter. A safe battery compartment will use some configuration of a locking mechanism to keep the battery secure and avoid the scenario where it falls out of the quadcopter. To enable a robot to replace batteries, it needs to posses the capability of unlocking and locking the protective lid of the battery slot, before and after it manipulates the battery itself. The robotic solution discussed in this report assumes that the battery does not contain a protective lid or a locking mechanism.

III. APPROACH

A. Setup

To develop, test and evaluate the system, a simplified adaptation of the battery replacement scenario was assembled. The imitation of a battery of dimensions 15 x 6 x 6 cm was 3D printed using PLA as the raw material. Similarly, a slot was also 3D printed to emulate the battery compartment of a quadcopter. To reproduce the variability of landing poses mentioned in subsection II-C, the slot was attached to a fixture with variable inclination and the ability to place it at any position on the ground plane.

A Franka Emika Panda Research Robot Arm is used as the robotic manipulator to physically interact with the 3D printed battery. To avoid complications, the battery was designed to have a protrusion for gripping which is compliant with the default end-effector attached to the robot arm. A camera is also fixed at a height near the robot arm to detect the various other components of the setup.

Lastly, an appropriately sized shelf was assembled to serve as the storage unit. The discharged battery will be stowed away in the top part of the shelf and the charged battery that is to be inserted in the slot as the replacement will be present in the bottom half.

To enable accurate detection of the battery, slot and the shelf, all of them were provided with ArUco markers. All of these components have been shown in Figure 1.

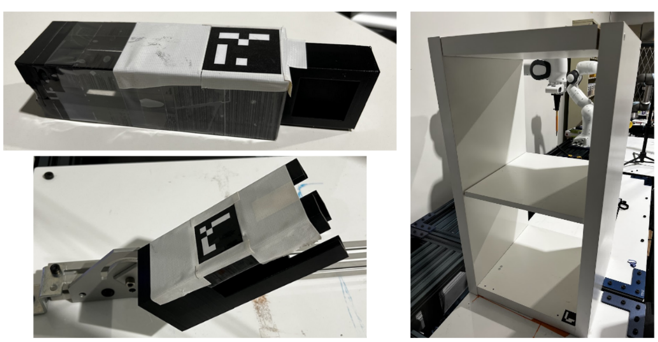

Fig. 1. Battery, slot and shelf

B. Extraction

After the quadruped has landed, the first step is to grasp and extract the battery from the slot. This involves detecting the battery, going to the grasp position, grasping the battery, pulling out the battery from the slot by following a linear trajectory.

C. Storage

Once the battery has been extracted and is in possession of the robot arm, it needs to be placed on the shelf for storage. This involves detecting the shelf, calculating the placement pose, moving to the placement pose and then successfully placing the battery in the designated area.

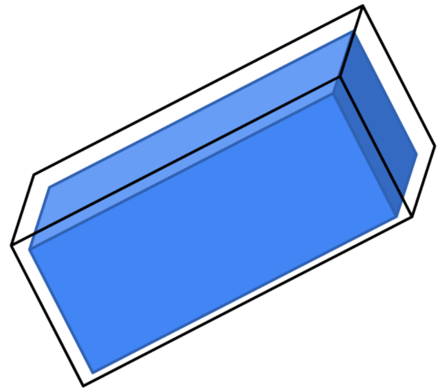

Fig. 2. Battery inserted into slot

D. Insertion

After the discharged battery has been stored and the new charged battery has been obtained, it needs to be carefully aligned with and inserted inside the slot. The final inserted configuration desired is shown in Figure 2. The blue cuboid represents the battery itself whereas the transparent cuboid represents the slot.

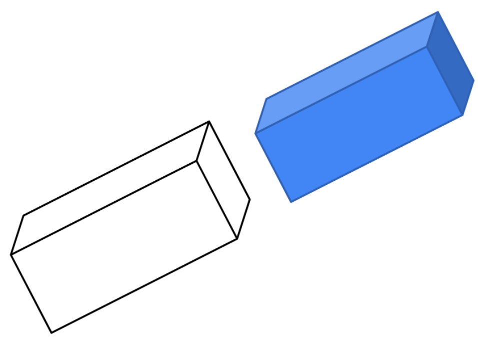

Fig. 3. Straight-fit alignment

1) Straight-fit: The simplest approach towards battery insertion is the Straight-fit approach. As visualized in Figure 3, the battery is aligned exactly along the axis of insertion, with the cross-sectional plane of the battery perfectly perpendicular to the cross-sectional plane of the slot. All the corners are aligned as well. In an ideal scenario, after this straight-fit alignment, the battery only has to be moved in the direction of the axis of insertion to achieve a successful insertion. Although, complications arise in this scenario as there will have to be low tolerances maintained along all the axes of the world frame throughout the insertion maneuver.

2) Edge-pivot: The second approach towards battery insertion alignment is the Edge-pivot, which is shown in Figure 4. At first, only the lower-back edge of the battery is aligned with the lower edge of the entrance of the slot. Once these two edges have been aligned, the battery has to be rotated about an axis which is collinear with the aligned edges until the cross-sectional planes of the battery and the slot are perfectly

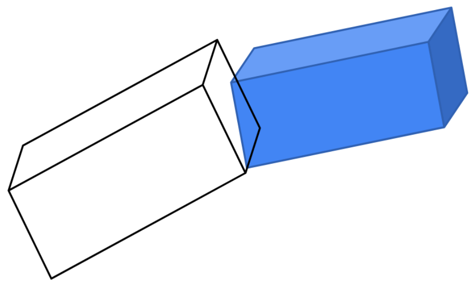

Fig. 4. Edge-pivot alignment

perpendicular. At this moment of the maneuver, a fraction of the battery is already inside the slot, which makes it easier to follow the axis of insertion without strictly monitoring the tolerances. In this scenario, only 2 of the axes of the world frame need to maintain low tolerances, while the third axis has higher tolerance.

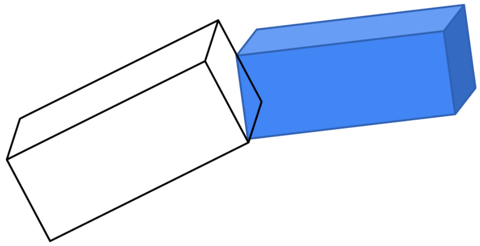

Fig. 5. Corner-pivot alignment

3) Corner-pivot: The final approach is the Corner-pivot, visualized in Figure 5. In this approach, initially only one of the corners of the lower edge of the battery is aligned with the corresponding corner of the slot. After this, the battery has to be rotated about an axis which coincides with the aligned corner, in such a way that the opposite corner is also aligned. Once this alignment is achieved, the battery then only has to follow the axis of insertion inside the slot. In this scenario, although there are higher tolerances along all the axes, the complication arises in rotating the battery after the corner alignment.

IV. Methods Implemented

A. Pose Estimation

To enable effective object tracking, we affixed distinct ArUco markers onto each object of interest. Specifically, we used ArUco markers ID 7, 33, and 9 to mark the shelf, battery, and slot, respectively, as shown in Fig. 6. The positions of these markers were then determined using an Azure Kinect camera.

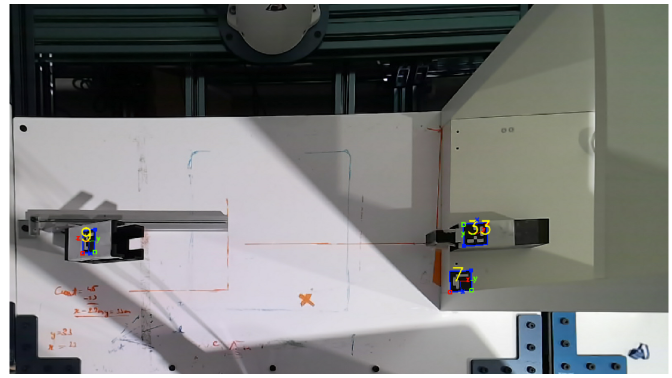

Fig. 6. ArUco Marker Setup

1) Camera Calibration: In order to utilize the position information derived from the ArUco markers detected by the camera, a conversion to the world frame is required. For this purpose, we performed camera calibration by employing a calibration block containing an ArUco marker with a known offset from the manipulator arm's end-effector. By utilizing forward kinematics, we were able to calculate the position of the ArUco marker in the world frame through multiplication with the known offset value. Given that the position of the ArUco marker was obtained in the camera frame, we subsequently determined the transformation matrix for the conversion to the world frame as follows:

$$T_c^w = (p_{ee_w} . p_{off}) p_{ar_c}^{-1} \tag{1}$$

where p_{ee_w} represents the position of the end-effector in the world frame, p_{off} denotes the known offset of the calibration ArUco marker in the world frame, and p_{ar_c} refers to the position of the ArUco marker in the camera frame. To mitigate the effects of measurement noise, a common approach is to obtain multiple samples and calculate the average result. In our study, we adopted this method by taking multiple measurements and averaging the results.

2) Pose Detection: To enable tracking of the ArUco markers fixed on the objects, we utilized the ROS package `aruco_ros`. Specifically, we developed a ROS node that subscribes to the topic published by `aruco_ros`, extracts the pose of the ArUco markers in the camera frame, differentiates the objects using their respective ArUco IDs, performs the conversion of the pose from the camera frame to the world frame using the transformation matrix, and finally publishes the pose information on distinct topics.

3) Data Filtering: The measured pose of the ArUco markers exhibited significant noise, particularly in the orientation component. To mitigate this issue, we applied a moving average filter with a window size of 20 to the raw pose measurements. This filtering method effectively stabilized the pose information obtained from the camera.

B. Task Planning

For the higher level task planning, we use a simple Finite State Machine shown in Fig. 7.

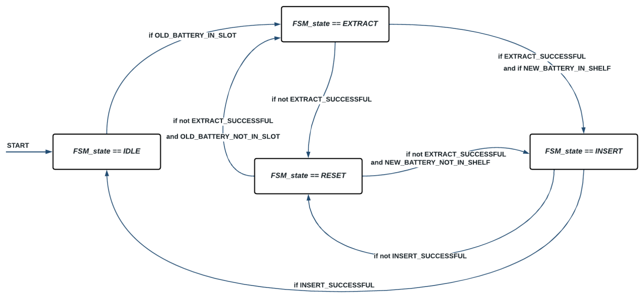

Fig. 7. FSM Task Planning

C. Motion Planning

Although the `frankapy` library provides some basic utility functions such as `goto_joints()` and `goto_pose()`, it lacks collision avoidance capabilities, and its `goto_pose()` inverse kinematics are not highly accurate, with errors of approximately 2-3 cm from the desired position.

To address these issues, we employed the `moveit_ros` package and its RRT* planner to plan the manipulator arm's motion from one point to another. Additionally, we utilized the `compute_cartesian_path()` function to move the end effector in a straight line, a crucial requirement for the process of battery insertion and removal from the slot.

D. Collision Environment

Using the `moveit_ros` package's capability to include collision objects, we defined several regions for the manipulator arm to avoid during the execution of its tasks. One of these objects was the shelf, whose dimensions we had previously measured, and whose position we had detected as discussed in Subsection IV-A. We also modeled the battery slot as a collision object. In addition, we added a plane 2 cm above the ground as another collision object. The battery itself was not included as a collision object, as doing so would have further complicated the tasks and potentially introduced problems during execution.

E. Battery Extraction

To extract the drained battery from the slot, the following steps are followed in a forward process:

1) Detect Battery Slot: As described in Subsection IV-A, we obtained the position and orientation of the battery slot by detecting the ArUco marker that we had affixed to it.

2) Calculate Desired Gripper Pose: We assume that the battery is always inserted in the slot with the ArUco marker facing the same direction. Thus, once we obtain the pose of the slot, we can estimate the pose of the battery gripping protrusion with a fixed offset. This offset is defined by the transformation matrix from the ArUco marker on the slot to the gripping position, which has been measured to be:

$$T_s^g = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0.08 \\ 1 & 0 & 0 & -0.06 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Note that this transformation should be performed in the ArUco pose frame. Therefore, we calculate the desired pose for the gripper in the world frame, denoted as p_{g_des}, using the following equation:

$$p_{g_des} = p_s T_s^g \tag{3}$$

where, p_s is the pose of the slot in the world frame.

3) Move to Grab Battery: To avoid any possible collision with the battery, a two-step maneuver is performed. Instead of directly moving to the calculated pose to grab the battery, the manipulator arm first moves to a position that is slightly outward from the final desired position, i.e., at a safe distance from the battery. The distance chosen for this step is 7 cm. The transformation required to move from the desired position to this new position is given as

$$T_{des}^{safe} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -0.07 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

When moving from the safe position to the desired gripping position, we utilized the `compute_cartesian_path()` function of the `moveit_ros` package to move the arm in a linear fashion.

4) Remove battery from slot: After gripping the battery by closing the gripper, we remove the battery from the slot by providing a new desired position for the manipulator to move to. This new pose is calculated in the same way as we calculated the safe distance while approaching the battery as discussed previously. The transformation to calculate this new position is given by the following expression:

$$T_{des}^{safe} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -0.20 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

We pull the battery 20 cm from the slot to ensure safety while executing the remaining tasks.

5) Place battery in shelf: As the shelf remains fixed during the operation, we pre-calculate the position where the extracted battery should be placed. Similar to the maneuver we performed to extract the battery, we perform this operation in two steps to prevent collisions between the battery and the shelf, especially since the shelf is not modeled in the collision environment.

First, we move the manipulator to a position that is 20 cm away from the target placement position and 2 cm above it. Then, we slide the battery onto the shelf and open the grippers to place it. It is important to note that we position the battery in a way that keeps the grippers in the vertical plane to prevent the battery from becoming stuck to them when they open. Once the battery is placed, the arm moves out from the shelf by approximately 10 cm.

F. Battery Insertion

To insert the new battery into the slot, the following steps are followed as a reset process:

1) Detect New Battery: We get the position of the new battery to be picked up using the pose estimation method discussed in Subsection IV-A.

2) Calculate Desired Gripper Pose: This process is exactly the same as the process (2) we described in the battery extraction process, with no other modifications.

3) Move to Grab Battery: Similar to step (3) of the battery extraction process, we use a two-step maneuver to approach the new battery in the exact same way. Once we have reached the desired position, we close the gripper to grab the new battery.

4) Move near Slot: After grasping the battery, we move the arm to a position that is 20 cm away from the slot and oriented towards it. To calculate this new position, we follow the same procedure as in process (2) of the battery extraction process, given by

$$T_s^i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0.2 \\ 1 & 0 & 0 & -0.06 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

5) Battery-Slot Alignment: During battery extraction, the battery and the slot were always aligned in the same orientation, allowing us to command the gripper to orient in the same direction to remove the battery. However, while grabbing a new battery from the shelf, the battery may not be perfectly aligned with the gripper. Therefore, during insertion, it is important to align the battery and the slot, rather than the gripper with the slot, to ensure successful placement.

To align the battery and the slot during insertion, we first calculate the desired pose for the ArUco marker on the battery, given the pose of the ArUco marker on the slot. The transformation required for this is given by the matrix T_{slot}^{bat}, which is a fixed transformation based on the known distance and orientation between the two markers, and is given as

$$T_{slot}^{bat} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.21 \\ 0 & 0 & 1 & -0.007 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

Using this transformation, we can obtain the desired position $p_{bat_{des}}$ for the ArUco marker on the battery, relative to the slot's coordinate frame using the equation

$$p_{bat\_des} = p_s T_{slot}^{bat} \tag{8}$$

However, the current position of the battery might not exactly match the desired position, and we need to calculate a transformation that will move the current position to the desired position. This transformation is given by T_{cur}^{des} and can be calculated by

$$p_{bat\_des} = T_{cur}^{des} p_{bat\_cur} \tag{9}$$

$$\implies T_{cur}^{des} = p_{bat\_des} p_{bat\_cur}^{-1} \tag{10}$$

Once we have the required transformation, we can apply it to the current position of the end effector to obtain the new desired position of the gripper. This new position is given by

$$p_{g\_des} = T_{cur}^{des} p_{g\_cur} \tag{11}$$

6) Insert Battery into Slot: After aligning the battery with a slot, the battery is inserted by pushing it in the direction of the slot. After insertion, the gripper is opened and the arm moves away from the battery. Next, the arm is moved to a reset position to prepare for the next battery extraction.

G. Reset Process

During operation, if the battery falls down and the task fails, a reset process is triggered. The task planner first checks if the battery has fallen during extraction or insertion. If the battery fell during extraction, the robot will pick it up and place it back on the shelf. If the battery fell during insertion, the robot will pick it up and place it back in the slot. After the battery is picked up, the robot will move to the reset position and the task planner will re-plan the task. This ensures that the robot can recover from a failed task and continue to operate effectively.

V. Evaluation

1) Success Rate: The system was evaluated by measuring the success rate of the extraction, storage and insertion maneuvers. Each of these maneuvers were conducted 15 times, out of which the successful and efficient executions were documented. The result for each of the maneuvers is listed in Table I.

TABLE I
Maneuvers Success Rates

Maneuver Type	**Maneuver Metrics**		
	Total Attempts	***Successful***	***Success Percentage***
Extraction	15	15	100
Storage	15	14	93.33
Insertion	15	11	73.33

2) Variability Achieved: As discussed in subsection II-C, one of the main challenges is the variability in the position and orientation of the battery slot. There are various configurations possible for the pose of the slot. To evaluate the implementation of the ArUco marker detection and capability to address the above mentioned variability, the system was tested with multiple orientations and positions. The slot has 3 degrees-of-freedom. The base of the slot was mounted on a single screw, so it is represented similar to cylindrical co-ordinates, by yaw and radius. The slot is mounted with another screw on the base, which allows for movement only in the pitch or inclination. All three of these degrees-of-freedom were varied for this test. The results of this test are mentioned in Table II.

3) Demonstration: A working video demonstration of the system is available here.

TABLE II
VARIABILITY ACCOMMODATION

Element	***Optimal Value***	***Operational Range***
Inclination	45 deg	±10 deg
Yaw	0 deg	±5 deg
Radius	5 cm	±3 cm

VI. FUTURE WORK

The present implementation of the autonomous battery replacement can extract a battery from the battery slot, store it within a shelf and insert another battery back into the slot. Although this implementation can successfully replace batteries in the setup described, the system can be made more reliable and accurate with a few modifications. Going forward, a camera placed on the manipulator end-effector itself will improve the system accuracy as that will grant the camera enhanced view of the objects, resulting in better tracking capabilities.

The system can also be improved by adding the means to detect and deal with a secure locking mechanism for the battery. This will more accurately represent a real-world application scenario to test and evaluate the success of the system.

ACKNOWLEDGMENT

We would like to thank Prof. Oliver Kroemer for his guidance, support and for giving us the opportunity to successfully execute this project on the Franka Robot Arm.

We also extend our thanks to Vibhakar Mohta from our cohort, for his contribution to the integration of `moveit_ros` package with the `frankapy` library. His contribution helped our implementation of the autonomous battery replacement solution.

REFERENCES

[1] Aurambout, JP., Gkoumas, K. & Ciuffo, B. Last mile delivery by drones: an estimation of viable market potential and access to citizens across European cities. Eur. Transp. Res. Rev. 11, 30 (2019). https://doi.org/10.1186/s12544-019-0368-2

[2] E. Frachtenberg, "Practical Drone Delivery," in Computer, vol. 52, no. 12, pp. 53-57, Dec. 2019, doi: 10.1109/MC.2019.2942290.

[3] D. Lee, J. Zhou and W. T. Lin, "Autonomous battery swapping system for quadcopter," 2015 International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 2015, pp. 118-124, doi: 10.1109/ICUAS.2015.7152282.