Mail Bot

Andrew Brooks, Carolyn Kim, Kevin Supakkul

Abstract

Automated driving is a rapidly growing field that will disrupt many industries in the near future by changing the way that we approach transportation. One industry that will likely be impacted is the delivery of mail and other packages. Our project's goal is to show one possible delivery solution that combines robotic manipulation with driverless car technology to create a fully automated mail delivery system. In the real world, one of the greatest challenges for this project would be developing a robot that is capable of handling the wide variety of packages and mailboxes that it would interact with during delivery. Our simulated robot addresses some of this variety by handling packages of various sizes and using force compliance to open a mailbox of unknown size. Our project also addresses the challenge of simultaneously controlling multiple subsystems with different controllers to fit each need. The final implementation of Mail Bot can successfully deliver packages (and animals) of various sizes to several mailboxes, using force compliance to open and close the box.

Final Implementation

Our robot is designed to accomplish all tasks that would be associated with mail delivery, including driving to the mailbox, opening the mailbox, grabbing a parcel, placing the parcel in the mailbox, closing the mailbox, and retracting the arm in preparation for the next mailbox. A state machine that outlines the desired tasks is shown in **Figure 1**.

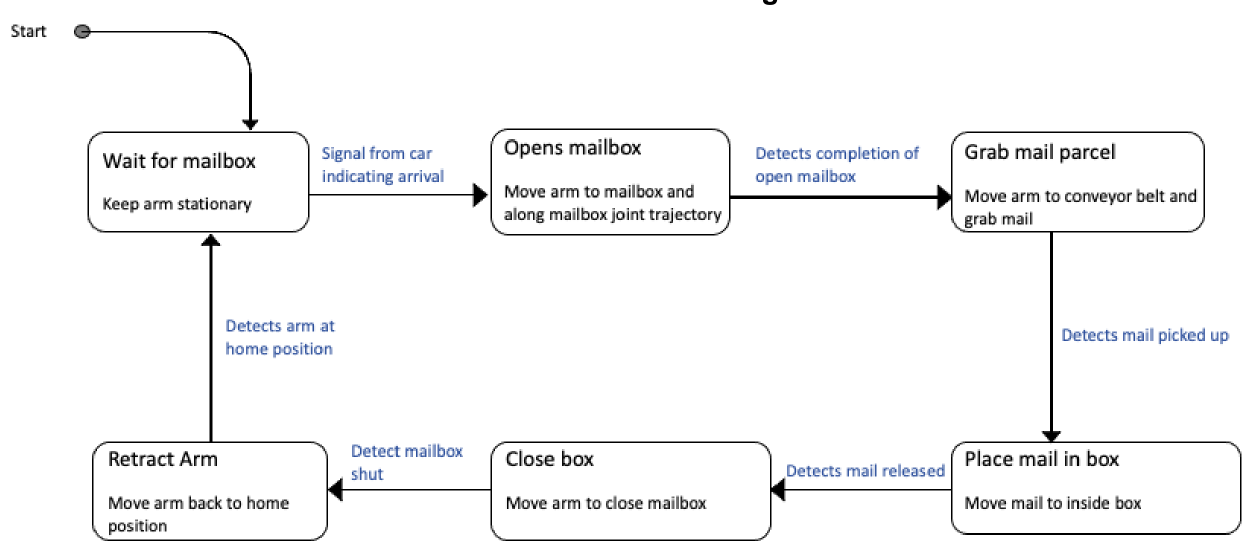

Figure 1 - Mail Bot state machine.

To accomplish all of these tasks we took a modular design approach, developing several robust controllers that could be used to accomplish similar tasks across all states. In the simulation environment the truck, arm, and gripper are combined into a single 12-DoF robot. Controlling each system separately required dissecting the mass-matrix, jacobian, and operation space matrices into smaller matrices that were specific to the joints of each subsystem. After breaking out each subsystem we were able to design a controller specifically suited to the needs of each one, which was especially necessary when controlling the robotic arm in operation space.

Both the truck and the gripper were able to use simple proportional derivative (PD) joint-space controllers to control their motion in each state. In our simulation the truck is treated as a simplified 1 degree of freedom prismatic joint that can only translate forwards and backwards. For this reason, its position is easily controlled with a PD controller in joint space, as we can set the desired truck joint position to equal the location of each mailbox. The joint space controller contains dynamic decoupling; by scaling the gains with the mass matrix, we are able to treat the controller as a unit mass system and choose gains to achieve critical damping. During truck motion, we also feed forward the coriolis, centrifugal, and gravity terms to compensate for dynamic effects and reduce error from the desired joint positions. The truck additionally uses velocity saturation control to prevent traveling at excessive speeds. In the real world, an automated truck would require a far more sophisticated control system, but this was not the primary focus of the project and this simple controller is sufficient for movement between mailboxes.

The gripper similarly uses a joint space PD controller. The only difference is that we did not feed forward coriolis and centrifugal terms for the gripper control, since it is unnecessary as our state machine dictates that all other joints remain static during gripper motion. A generalized PD control equation that represents the control used on the gripper and truck is shown in **Equation 1**.

$$\Gamma = A(-k_p(q - q_d) - k_v\dot{q}) + b + g$$

Equation 1 - PD control equation for truck and gripper

The arm requires a more sophisticated position-orientation controller in operational-space. In order to accomplish all of the tasks related to delivering mail from the truck to the mailbox the arm must be able to reach known locations with a specific end-effector orientation. In our simulation the robot grabs packages from a known location inside the truck and interacts with a mailbox, which would be detected through computer vision in a real world system. Accomplishing complex delivery tasks requires the robot to have redundancy, so there are more degrees of freedom than are actually needed to accomplish each position and orientation. To deal with redundancy, we control posture in the null-space, using one of two known task-specific postures that we select depending on whether the robot is grabbing mail inside the truck or interacting with the mailbox. The arm requires dynamic decoupling to allow us to treat the controller as a unit mass system and choose gains to achieve critical damping. Additionally, it uses velocity saturation at the end-effector to ensure safe movement of packages.

Opening and closing the mailbox requires a slight modification to the arm controller to allow for compliance in a specified direction. To achieve this a selection matrix, Σ_F, is added to the linear force control. This selection matrix can zero out forces along a specific axis to achieve compliance and will be an identity matrix if no compliance is necessary. When opening the mailbox, the robot pulls horizontally on the handle and maintains vertical compliance to smoothly follow the radius of the mailbox door as it opens, as shown in Figure 2. Alternatively, when closing the mailbox the robot pushes vertically and maintains compliance in the horizontal direction.

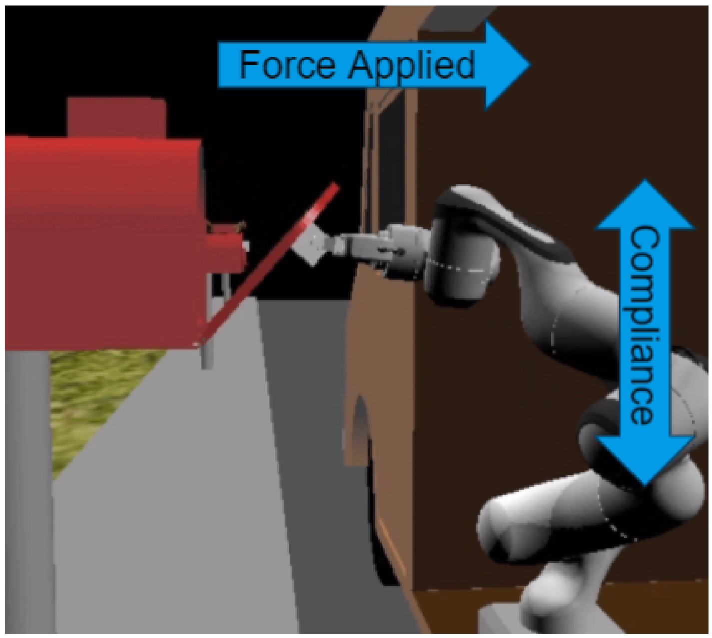

Figure 2 - scheme for compliant opening of mailbox

The equations used to implement the arm controller are shown below in **Equation 2**, with the final command torque matrix also including the control equations for the truck and gripper joints, which must maintain their positions during arm movement.

$$\dot{x}_d = \frac{k_p}{k_v}(x_d - x)$$

$$v = sat\left(\frac{V_{max}}{|\dot{x}_d|}\right)$$

$$F = \Lambda_0 \begin{bmatrix} \Sigma_F(-k_v)(\dot{x} - v\dot{x}_d) \\ -k_p(\partial\phi) - k_v\omega \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} M_{truck}(-k_p(q_{truck} - q_{d_{truck}}) - k_v\dot{q}_{truck}) \\ J_{arm}^T F + N^T M_{arm}(-k_p(q_{arm} - q_{d_{arm}}) - k_v\dot{q}_{arm}) \\ M_{gripper}(-k_p(q_{gripper} - q_{d_{gripper}}) - k_v\dot{q}_{gripper}) \end{bmatrix} + g$$

Equation 2 - Operational-space position-orientation control equation for the arm. Note that when controlling the arm in operational space, we also control the truck and gripper joints using PD control to maintain their desired joint positions.

The combination of these controllers to control each of the subsystems of our robot allows us to achieve all tasks in a basic automated mail delivery simulation.

Challenges

During this project we encountered challenges related to both control and simulation. The simulation challenges primarily involved implementing collision-objects in the SAI2 simulation environment. In the real world, a package will interact with different surfaces during the delivery process, in the truck, at the gripper, and inside the mailbox. Most of these collision surfaces were successfully modeled in SAI2, but adding additional collision objects inside the truck proved very challenging since the truck is part of the robot in simulation. Additionally holding the mail parcels in a container during travel would cause continuous collisions between several objects, which would slow down the simulation. To avoid these problems, the mail parcel positions are locked relative to the truck until gripped, and the dynamics of each parcel are enabled after gripping. This simplification reduced the computation in our simulation and allowed us to focus on more significant control challenges.

In our controller, the first large challenge that we faced was controlling the subsystems of the robot separately. The truck, arm, and gripper must move independently of one another in order to achieve each task efficiently. As stated earlier, the solution to this challenge involved dissecting the mass-matrix, jacobian, and operation space matrices into smaller matrices that were specific to the joints of each subsystem. Doing this required us to write our own controllers from scratch for each system, since the provided controllers are meant to control all joints together. However, once these systems were separated it allowed us to set a specific task for each subsystem in each state. This greatly improved the modularity of our controllers and allowed us to apply the same controllers across various tasks.

Another control challenge was opening and closing the mailbox. The simplest way to achieve this task would be to program the controller to follow a known radius of the mailbox door, but this solution would be counter to our goal of working with a variety of mailboxes. To work with any mailbox door, we needed to use force compliance. The challenge in maintaining compliance is that the end-effector frame is constantly changing orientation if it stays locked with respect to the mailbox handle. To simplify this problem, we implemented a rotating collision mesh at the mailbox handle, which allowed the robot to maintain orientation while opening / closing the mailbox. With a consistent orientation, we were able to zero forces in a single direction to achieve basic compliance that was sufficient for opening a mailbox with a handle-to-hinge radius of unknown size.

If we were to start the project again, our initial approach would be the same. Breaking the controller out into separate subsystems was a crucial step that made the implementation of each state more manageable and easier to comprehend. The control structure that we developed is easy to build upon and add additional states and features. But with more time, we would improve our controllers and add more functionality. Our current system does not incorporate force feedback for controlling the grippers; we utilize a high gain to maintain grip of the mail, but we would implement force feedback to prevent crushing of more fragile packages. We would also implement computer vision to detect mailboxes with different types of hinges (vertical or horizontal), and we would scale our system to deliver multiple parcels of mail to each mailbox by utilizing a wifi-enabled canister to actuate several letters of varying sizes. And finally to make our mail delivery system more robust against external disturbances, we would incorporate negative potential with computer vision.

Conclusion

This project was a perfect capstone to our study of robotic control. We chose a project and set a project scope that seemed feasible to implement in a short period of time, but we quickly saw how many unexpected challenges can arise in even simple control problems. Breaking down our robot joints and writing separate controllers for each allowed us to explore the range of controller design that can be used for different applications, from simple PD control for robots with few degrees of freedom to complex position-orientation controllers with null-space posturing to accommodate robots with many degrees of freedom. An important lesson from this project was the importance of understanding your control algorithm and not being afraid of starting from scratch to simplify the controller. We have learned a lot about different control algorithms throughout this course and appreciate the importance of understanding each one and when to use it.