Project proposal

EECS 452
Winter 2023

Intuitive Control of a Robotic Arm

Justin Yu, Jack Getty, Nick Farid, James Wishart

I. Introduction and overview of project

Human teleoperation of robots is an effective method to augment human capabilities in applications where the site is inaccessible, remote, hazardous, requires higher physical loads or higher precision than humans can provide alone, for example in underwater and space exploration, mining toxic materials, and surgery. Various teleoperation control schemes have been employed for these applications.

We distinguish between two primary forms of robotic teleoperation. The first involves a tactile interface where the robot is controlled by a device that the human user contacts directly, like joysticks, gamepads, and haptic devices, which can receive human bio-signals passively. The second is contactless control, where the robot can be teleoperated by signals through vision or audio inputs or otherwise.

For the first form especially, the mismatch between the user control space and the robot control space can increase the difficulty of telemanipulation and lead to poor operation. Development of more intuitive control schemes can lead to a decrease in the mental load required from the operator and lead to higher efficiency and effectiveness in robot tele-operation. In the future, a fusion of these two forms may be deployed, combining the two techniques to achieve contactless control with feedback, possibly involving mixed-reality (MR) devices and the use of brain-machine interfaces or machine-nervous system interfaces.

For our project, we aim to create an intuitive real-time teleoperation control scheme for robotic arms that emulates the movement of a human arm-hand effector system, falling under the second form as previously described. Stereo Computer Vision (CV) with an integrated Machine Learning (ML) pipeline will be used to implement hand pose estimation and localization in 3D space. The localized hand pose is re-mapped to the coordinate frame for a 6 DoF RA. The joint angles are calculated and the robot arm is actuated to achieve the desired pose.

There are several challenges that we will encounter throughout the development of this project. Several problems arise from the fact that a human arm-hand system is largely not analogous to a 6DoF robotic arm. The linkage structures and dimensions are inherently different. For example it is not sufficient to track the arm pose and send the inner-elbow angle as robot actuation commands. Due to this, the Inverse Kinematics (IK) solution for the arm is required to solve for robot joint angle solutions using only information about a desired end-effector pose. We also have to appropriately scale the remapping of the real-world frame hand pose representation to the RA frame. A one-to-one distance mapping will likely not be sufficient and may result in several instances of out-of-bounds conditions in the robot configuration space during operation. Robotic arm self-collision must also be reconciled. There are several naive implementations to achieve this but there are also more effective solutions that we can explore, time allowing.

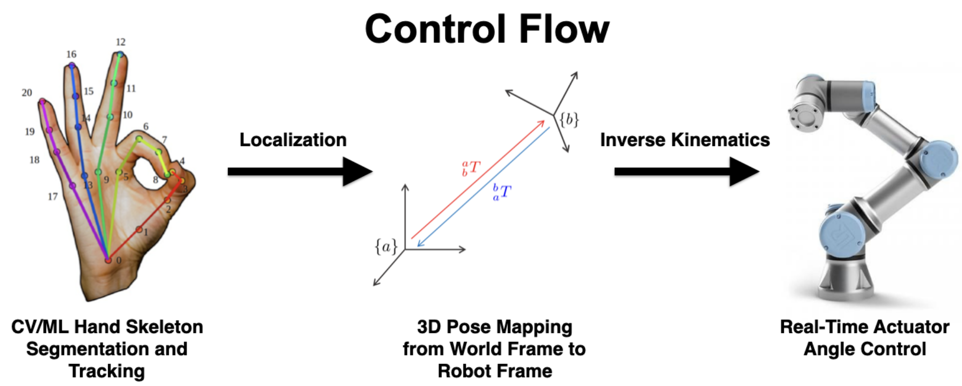
For hardware we require stereo cameras, a Raspberry Pi to serve as the core computational system and process camera inputs, and a Teensy to take our data and generate PWM signals for servo actuator control of the 6 DoF RA.

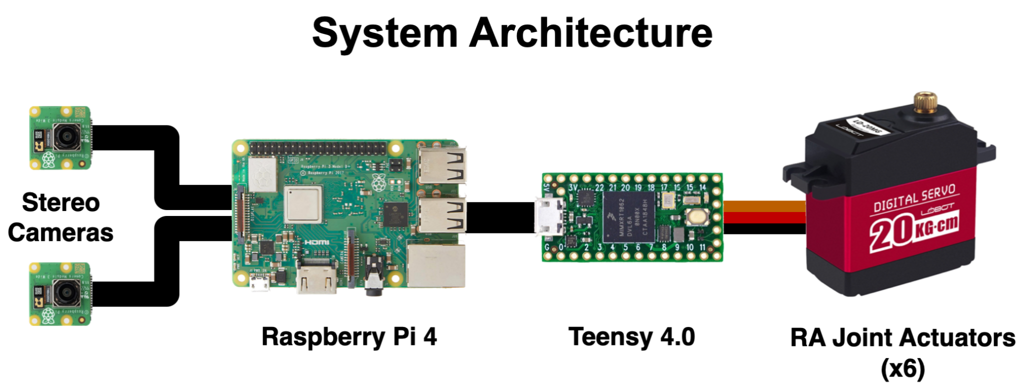
We intend to demonstrate this system with recorded and hopefully live demos.

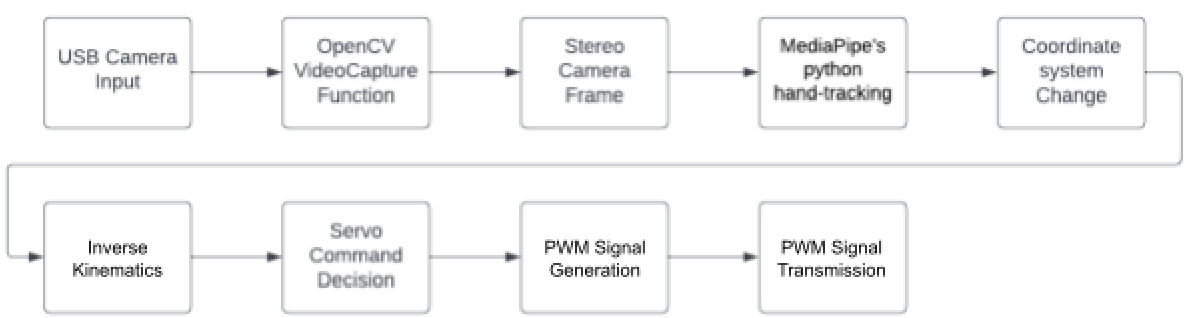
Materials:

1. 2 sheets of acrylic or wood with dimensions 30 cm * 50 cm
2. 6 servos
3. Teensy
4. Raspberry Pi
5. Stereo Camera
6. Arduino IDE
7. Python IDE for the PI
8. Resistors of values unknown at the point (for serial communication)
9. Wires (many)
10. External power source for boards and servos (5V)
11. Screws (3mm)
12. Breadboard
13. *Servo controlled gripper (for stretch goal)

II. Description of project







System Flow Block Diagram

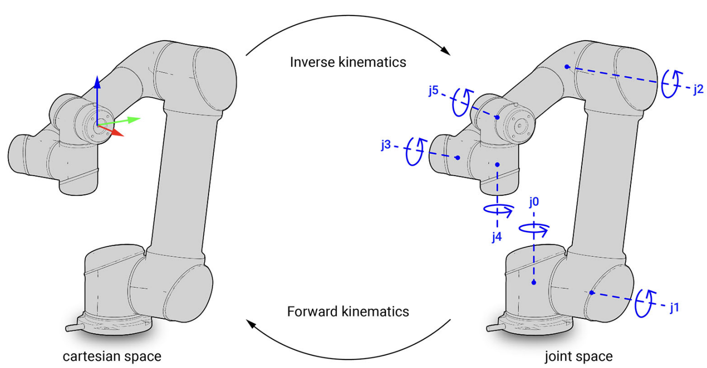
- The stereo cameras serve as our control system's visual input to track hand pose in real time. We opted for stereo cameras instead of a single (mono) camera system because we intend to achieve a 3D representation of the hand skeleton pose. Stereo cameras will allow us to gain depth information to build the desired representation. The data will be sent to the Pi over USB

- The Pi acts as our main signal processor. The images will be captured on the Raspberry Pi, where it then undergoes processing with CV/ML techniques using MediaPipe's python hand-tracking library. After the hand is identified and localized, we will transfer the hand coordinates to the robot frame, then translate those coordinate changes into motor commands that will be sent to the teensy. Communication between the Pi and Teensy will be done with UART as we only need two devices on the bus, only one needs to send any information, and UART has an existing library for use on both the Raspberry Pi and teensy.

- The teensy acts as our motor controller. It will accept packets from the Pi over UART that correspond to different servo movements. These movements will then be sent along to the robot arm via PWM waveforms.

- The Robot arm will be the final output. Its servos will be controlled with PWMs with movements dictated by the teensy.

Algorithms/Techniques



- **Forward/Inverse Kinematics (FK/IK):** The Forward Kinematics problem is concerned with the relationship between the individual joints of a robot manipulator and the position and orientation of the tool or end-effector. Formally, the solution to the forward kinematics problem is the determination of the position and orientation (the pose) of the end-effector, given joint variable values of the robot. The joint variables are the angles between the linkages in the case of revolute (rotational) joints, and the linkage extension displacement in the case of prismatic (sliding) joints. The forward kinematics problem is to be contrasted with the inverse kinematics problem, which is concerned with the determination of values for the joint variables that achieve a desired position and orientation for the end-effector of the robot. To achieve the kinematic solution for our robot, we will employ the Denavit-Hartenberg frame convention.

- **Stereo Image Triangulation/3D Hand Pose Mapping:** In order to understand where our hand is in three dimensions we will need to use the information we know about the separation of the cameras and differences in the coordinates received from each to estimate the depth in the image. This can be done in OpenCV. By finding the disparity in the measurements from the two cameras, then relating them to the intrinsic parameters of the camera we can obtain a depth map. This equation works out as $depth = \frac{Bf}{disparity}$ where B is the baseline of the stereo camera and f is the focal length.

- **Gram-Schmidt Orthogonalization Algorithm:** Information about a pose in 3D space is fully defined by 3 linear displacement and 3 rotational displacement arguments (rotation along orthogonal basis axes). A common way to represent pose information is by using an orthonormal frame in $\mathbb{R}^3$, similar to how the origin frame is represented for a 3D coordinate system. The Inverse Kinematics problem as previously described requires a pose as the input. However, we cannot directly achieve pose information from the 3D hand skeleton representation that we intend to develop. For this, we will likely use the index finger direction to define the primary frame axis, and use the thumb to define our secondary frame axis. We however cannot guarantee the index finger and thumb to be orthogonal, so we will allow the index finger and thumb to define a plane, and establish the third axis in the plane normal direction. To build an orthonormal frame from two non-orthogonal axes we require the Gram-Schmidt Orthogonalization algorithm, which involves projecting the desired axes onto normal planes to achieve an orthonormal frame.

- This project will assume that lighting conditions are relatively fair, that is to say that the room is well lit and light is not shining directly into the camera itself. If this were not true the camera would have a much harder time accurately tracking the user if it can do it at all. We will also assume that the user will not be moving their arm excessively quickly but rather in slower smooth movements. This will also make it easier for us to track the arm and will make our real time deadlines slightly more forgiving as the robot will not need to react as quickly.

III. Milestones

- Milestone 1: Design Milestone
  For Milestone 1, we would like to have a fully developed robotic arm idea and finish the prototypes of the functions we will be designing. For the arm we would like parts to be ready to be ordered by the milestone checkpoint, as well as any manufacturing files such as the support beams and base created in CAD so that we will be able to manufacture

them immediately following the milestone review. For the computer hardware we will have selected a serial interface to communicate between the pi and the arduino, we will have selected an appropriate stereo camera for our purposes, we will know power requirements and adapters necessary. We will also have wiring mostly planned out. A bill of materials should be ready to purchase any items, or collect the items we have on hand. On the code aspect, we will be looking to write function prototypes for both the pi and necessary components. This includes the code for the stereo camera, the filtering system we need for hand detection and the control system for the servo control. We are not aiming to have flushed code at this point, just the method and overarching plan of each subsystem.

- Milestone 2: Product Milestone
  For milestone two we will have the arm completely constructed and able to be moved. That is to say movements of an arm in front of the camera will cause the arm to move. However we do not expect that the system will be fine tuned at this point. The arm may still move unexpectedly or incorrectly, and be impacted by lighting conditions of the room. In addition we do not expect that we will have a gripper at this point but this is the latest we would decide to add one if there is time in the project. As a fallback, if stereo vision is not working we may move to mono-vision and movement in a single plane. If we are having trouble with mimicking specific arm movements we may move to certain gestures meaning certain movements. Depending on the status of our project, we will decide if we will implement the gripper prior to milestone two in order to allow us to have enough time for extensive testing and debugging.

- Between Milestone 2 and our final report we will be aiming to complete extensive testing and debugging. Our system should be mostly done by the end of Milestone 2 to allow this testing window. This will allow us to get closer to our goal of an accurate and lag-free system.

- Final Deliverables:
  For our final deliverable we want to be able to detect and follow a hand in 3D space using a stereo camera. Real-time processing of the images will be done on a raspberry pi and converted into real coordinates either directly controlling the servos in the arm, or being sent over serial communication to a microcontroller that will move the arm. The arm itself will allow for 6 degrees of freedom to allow movement to anywhere in 3D space.

IV. Collaboration plan and team member contributions

We plan to divide our responsibilities into two primary subsets. The first set of responsibilities will be focused on the hand segmentation and kinematics process, which will be substantially more software based. The second set of responsibilities will be related to the construction and design of the robotic arm which will be much more hardware based.

James and Justin will be mainly responsible for developing the hand segmentation algorithm and the arm kinematic model. This is because these two group members have the most experience with embedded software development, Justin through MRover and James through 473. This should enable them to quickly prototype on the PI without needing as much time to become familiar with it

Specific tasks within this section include

- Taking in stereo camera data
- Identifying and localizing the hand and arm in the frames
- translating hand and arm movements into serial packets corresponding to servo commands

Nick and Jack will be mainly responsible for developing the hardware related to the robotic arm. This is because they have the most experience with electrical subsystems as both are Electrical Engineering Majors. This should enable them to quickly begin fabricating and building once decisions on materials are made.

Specific tasks within this section include

- Modeling the arm pieces and deciding their material
- Fabing and assembling the pieces into the finished arm
- Wiring the servos and power delivery for the system

Once these two halves are done, we will move toward integration which will be the responsibility of everyone since Justin and James will have the most knowledge of the software expectations and Jack and Nick will have the most knowledge of the hardware expectations.

That being said, every group member will contribute to every part of our project. We plan to allocate jobs based on our collective progress. We plan to direct our focus towards problem areas accordingly.

In order to keep track of progress we will have weekly meetings either in the lab or over zoom to discuss what is working and what is not going well. From these meetings we will develop plans for progress during the next week. To securely store our code and have proper version control we have created a github repository where everything will be stored and every member will push their changes.

As a group we discussed possible conflicts throughout the semester.

All members anticipate being mostly gone and not working over the Spring Break, but we will still maintain communication and continue planning throughout the semester.

James and Nick should have none other than break that will force a reduction in work

Jack and Justin are each part of a major student project design team, Solar Car and MRover respectively. However, neither anticipates that there will be time conflict between their obligations to the project team and this project. If there is a conflict it should be ok as they each come from a separate subteam so progress on any one part of the project should not halt.

V. References and prior work

https://www.instructables.com/DIY-Robotic-Arm/ Refer to sections 2-4

The site explains an implementation of DIY Robotic arm, which also incorporates the use of servo motors and a servo controlled gripper. This project helped us with our ideation, particularly with how we plan to go about constructing the actual arm. This project differs from ours in the sense that it uses an arduino uno to control the servo motors and the inputs are controlled by a remote controller as opposed to a user’s hand gestures. We took inspiration from this project because we plan to incorporate servo motors in a similar configuration in order to create the joints of the arm. In addition to this, we plan to use similar 3D printing methods to construct the framing and connection pieces in the arm.

https://doi.org/10.1186/s10033-022-00813-1 Refer to sections 1-2

This paper helped with our ideation and motivated us to pursue the creation of a 6DoF robot because it enables improved robot motion performance and increases operation efficiency. Figure 1 in section 2 includes a system diagram illustrating the signal processing methodology and control framework, which will help us develop these elements in our project.

https://doi.org/10.3390/app12094740 Refer to section 1

This paper defines the inherent challenges that come from using 2D displays to represent the visual feedback from the robot. These concepts have helped us to understand that robot control is not intuitive and natural to users, which is why we opted to control the robot in the most seamless way possible by using hand commands and computer vision.

https://medium.com/@nicolas.gorrity/robotic-hand-gesture-control-using-machine-learning-and-image-processing-74b83b0da51a

This article provides a thorough breakdown of how we can extract hand features using image processing. The general steps are to use image processing to extract hand features, then the hand gestures are recognized using machine learning and finally the robot is manipulated to move accordingly. This link is valuable as it explains the use of fourier analysis to identify the hand shapes that are being made.

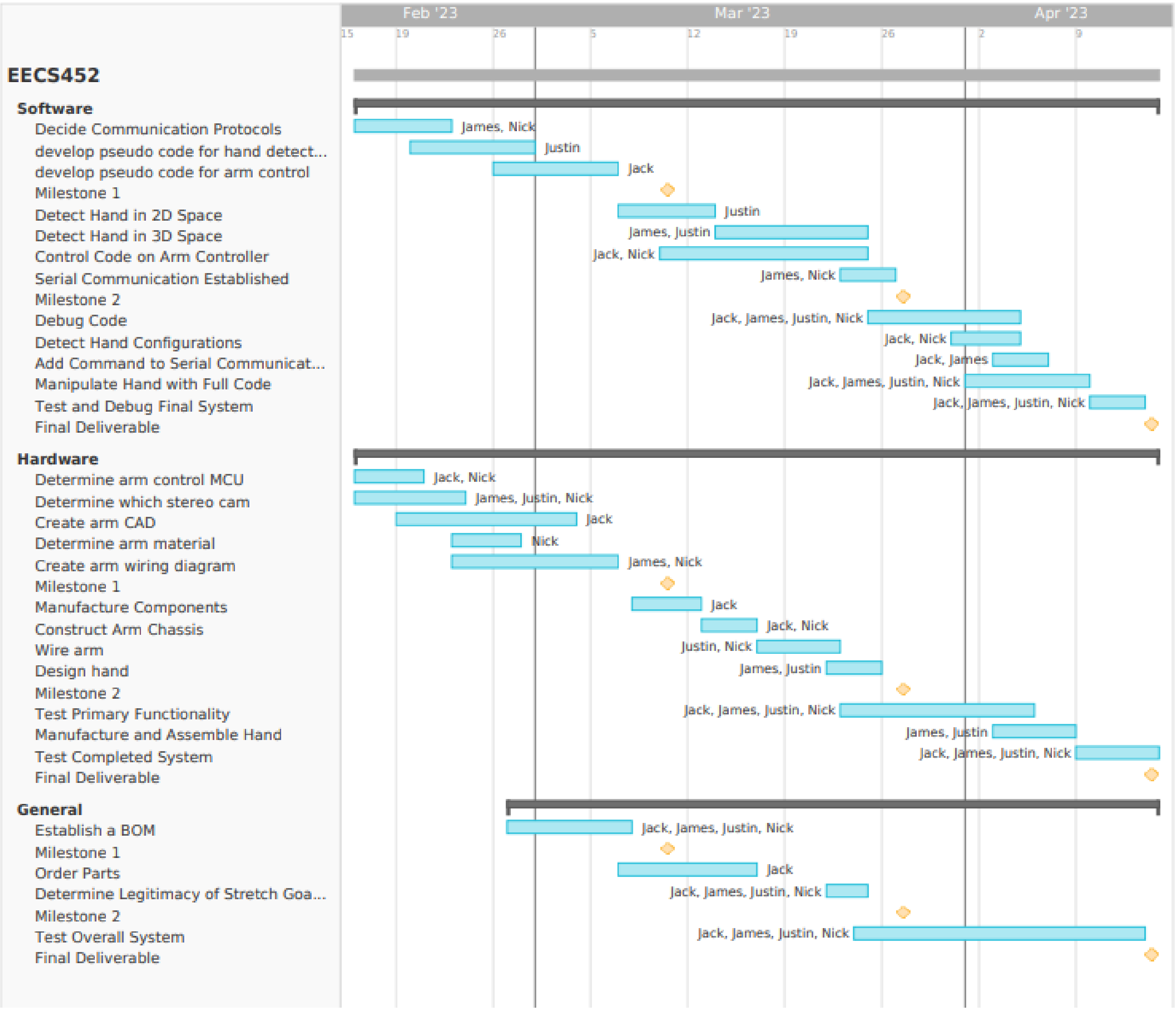
https://www.youtube.com/watch?v=9iEPzbG-xLE

This video provides a visual representation of our hand segmentation process, which we plan to replicate during our project.

VI. Project schedule and risk management

We created a gantt chart to keep track of our tasks and ensure any dependencies are addressed before each task is scheduled. Each task is given enough spacing between its dependencies to allow for other problems to arise and be solved before there is a hindrance. The big developmental stops are waiting for parts to arrive after ordering, the construction of the chassis as well as being able to detect a hand in 3D space. Each of these tasks have multiple moving parts that can cause delay, or the timing is out of our control on some levels. To ensure there will be no prolonged delays, we will order our BOM as soon as possible, preferably before Milestone One, and prioritize our main tasks to allow all members and subteams the ability to run smoothly and save enough time for overall testing at the end of the semester.

In the event that we are unable to complete one of these priority tasks, or parts are delayed, we do have a fallback plan. We believe that controlling the hand in 2D space is an adequate route to pursue if we are behind. This will remove the complexity of 3D detection and controlling the arm in 3D space. On the other hand, if we are ahead of time, our stretch goal is to add a manipulator to the end of the robotic arm. This claw will be able to mimic the controlling hand when it gestures an open palm versus a closed fist. Ideally this will allow our arm to interact with the world around it and lead to a more viable project. The stretch goal landmarks are included in the Gantt chart as well as a period to decide the final outcome of our project before the Milestone 2 deadline.


Feb '23
Mar '23
Apr '23
15
19
26
5
12
19
26
2
9
EECS452
Software
Decide Communication Protocols
develop pseudo code for hand detect...
develop pseudo code for arm control
Milestone 1
Detect Hand in 2D Space
Detect Hand in 3D Space
Control Code on Arm Controller
Serial Communication Established
Milestone 2
Debug Code
Detect Hand Configurations
Add Command to Serial Communicat...
Manipulate Hand with Full Code
Test and Debug Final System
Final Deliverable
Hardware
Determine arm control MCU
Determine which stereo cam
Create arm CAD
Determine arm material
Create arm wiring diagram
Milestone 1
Manufacture Components
Construct Arm Chassis
Wire arm
Design hand
Milestone 2
Test Primary Functionality
Manufacture and Assemble Hand
Test Completed System
Final Deliverable
General
Establish a BOM
Milestone 1
Order Parts
Determine Legitimacy of Stretch Goa...
Milestone 2
Test Overall System
Final Deliverable
James, Nick
Justin
Jack
Justin
James, Justin
Jack, Nick
James, Nick
Jack, James, Justin, Nick
Jack, Nick
Jack, James
Jack, James, Justin, Nick
Jack, James, Justin, Nick
Jack, Nick
James, Justin, Nick
Jack
Nick
James, Nick
Jack
Jack, Nick
Justin, Nick
James, Justin
Jack, James, Justin, Nick
James, Justin
Jack, James, Justin, Nick
Jack, James, Justin, Nick
Jack
Jack, James, Justin, Nick
Jack, James, Justin, Nick