Manipulator-based Haptic Controller for Car-like Vehicles

ME201 Final Project: Hyung Ju Suh (Terry), Peter Renn, Neha Sunil, Qifan Wang

I. Introduction

From its beginning, car-like vehicles with steering linkages have utilized steering wheels and a throttle pedal control scheme due to the ease of mechanical construction, and wide range of control. In the early designs the steering wheel was directly linked to the steering linkage through a kinematic geartrain. The more modern designs are hydraulically assisted, and most cars nowadays utilize electric-assisted power steering and a pedal for its control scheme.

However, in the context of teleoperation, the steering-wheel-and-pedal control scheme is often cumbersome to implement and transport. Additionally, since both the steering wheel and the throttle pedal is fundamentally limited to a single-axis feedback, they often cannot provide complete feedback regarding the car's dynamic state. Advanced simulator/teleoperation devices utilize a full platform movement to overcome this limitation, but such devices are extremely costly to implement and manufacture.

This work explores an attempt to utilize joystick-like driving schemes for teleoperating car-like vehicles, using haptic manipulators that are not only able to take encoder-based inputs, but also give force feedback based on torque. Utilizing manipulators for haptic feedback also allows us to control the force using various options for basis spaces, such as Cartesian or cylindrical, which enables us to effectively simulate the vehicle's dynamic state in a versatile way.

Past works have involved using manipulators as haptic controllers for car-like vehicles [1], mobile holonomic robots [2], and other manipulators such as excavators [3]. The latter often utilizes a master-slave method using haptic feedback. In the former, however, many interesting problems arise in how to map the vehicle's kinematic and dynamic states to the manipulator's joint-torque space. In our work, we mainly look at contribution from the accelerometer and obstacle sensing as factors that would feed back to torque applied to the manipulator.

Finally, we focus on the safety-aspect of vehicle control for teleoperated systems. With the feedback supplied from vehicle's dynamic states, it should be required that when a human lets go of the manipulator, the manipulator should home itself to a position where zero-velocity is commanded. This is passively true for a car's steering schemes as well, as the steering wheel and the throttle pedal would 'zero' itself without external human input.

II. Methods

A. Problem Statement

We first define the problem mathematically in the following way: we have a GPS-denied (indoor) environment where the car has on-board Inertial Measurement Unit (IMU) and collision-avoidance (distance) sensors. It receives teleop-commands through a manipulator-based controller, for which we can control the motor torques, with closed-loop joint position feedback. Then, the problem is to come up with a vehicle-velocity mapping $\vec{v} = f_v(\vec{q}, \vec{a}, \vec{\omega}, \vec{d})$, and a manipulator-torque mapping $\vec{\tau} = f_\tau(\vec{q}, \vec{a}, \vec{\omega}, \vec{d})$ which enables the user to teleoperate the vehicle with haptic feedback. The overall system diagram is illustrated in Fig.1.

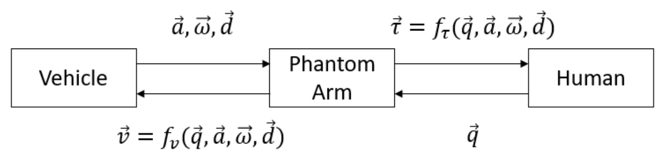

Fig. 1. Functions to design for a haptic controller with force feedback

Some of the interesting requirements for this tele-operated system is as follows:

1) Safe: if the human lets go of the controller, can it automatically go to a position where it will command zero velocity to the car?
2) Intuitive: is the controller intuitive to drive?
3) Feedback: What are the benefits of "feeling" the response out of the car?

B. Vehicle Dynamics

Traditionally, the vehicle lives in a two-dimensional planar space in $SE(2)$, with an added degree of freedom due to the steering angle. The traditional control for car-like vehicles is done through two control inputs: the first input controls the forward velocity of the car, and the second input controls the steering angle. In this case, we define the hybrid body velocity as a twist in $SE(2)$

$$\mathbf{V}^b = \begin{bmatrix} v_x^b \\ v_y^b \\ \omega^b \end{bmatrix} = \begin{bmatrix} u_x \\ 0 \\ \frac{u_x}{L} \tan u_\phi \end{bmatrix} \tag{1}$$

where L corresponds to the wheelbase of the car, and ϕ is the steering angle [4]. The Cartesian components are connected to the accelerometer and gyroscopes by

$$fracddt \begin{bmatrix} v_x^b \\ v_y^b \end{bmatrix} = \begin{bmatrix} a_x^b + \omega_z^b v_y^b \\ a_y^b - \omega_z^b v_x^b \end{bmatrix} \tag{2}$$

Note that we don't have a physical estimation of the body velocity, and in order to estimate true body velocity, we can estimate the magnitude of it by integrating acceleration and incorporating the Coriolis term by using data from the gyroscope. If we have the body velocity, we can estimate the full world-state of the vehicle using

$$\mathbf{X} = \begin{bmatrix} x^w \\ y^w \\ \theta \end{bmatrix} \qquad \frac{d\mathbf{X}^w}{dt} = \begin{bmatrix} \mathbf{R}_b^w & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{V}^b \tag{3}$$

The coordinate system of the vehicle is illustrated in Fig.2.

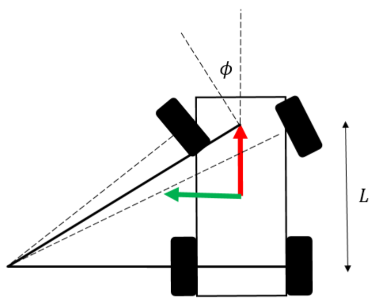

Fig. 2. Coordinate definition for the vehicle. We utilize XYZ to RGB convention.

C. Phantom Arm Dynamics

For the tele-op controller, we use a Phantom manipulator with the setup illustrated in Fig.3. The forward kinematics of this arm (from joint angles to Cartesian position of the end-effector) is illustrated in Appendix 1, as well as the Jacobian that maps the joint velocities to the Cartesian velocities:

$$\vec{x} = f_{kin}(\vec{\theta}) \qquad \dot{\vec{x}} = \mathbf{J}\dot{\vec{\theta}} \tag{4}$$

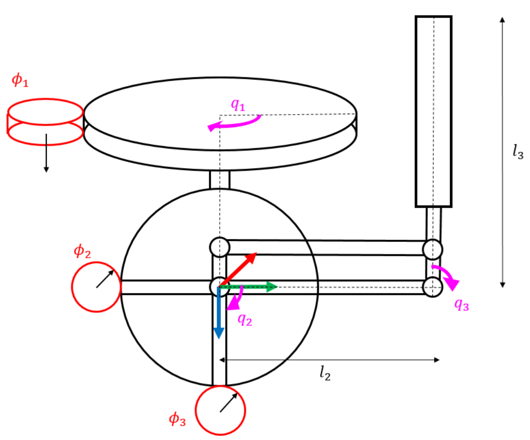

Fig. 3. Coordinate definition for the phantom manipulator. Again we utilize XYZ to RGB convention.

D. Phantom Arm Mapping

1) Joint Space Mapping: First we define the joint space of the phantom, denoted by relative degrees of freedom in a serial mechanism: $\vec{q} = [q_1, q_2, q_3]^T$. It makes intuitive sense for the yaw joint, q_1, to control the steering angle of the car, ϕ, since it allows for a circular planar motion that approximates the steering linkage of the car. After this, we map the x-direction to the final angle q_3, such that we have the following linear mapping:

$$\begin{bmatrix} u_x \\ u_\phi \end{bmatrix} = \begin{bmatrix} -C_x & 0 \\ 0 & -C_\phi \end{bmatrix} \begin{bmatrix} q_1 \\ q_3 \end{bmatrix} \tag{5}$$

2) Cylindrical Space Mapping: Another possibility is to make a cylindrical representation of the whole space, which abstracts the second and third joint angles to distance from axis. In order to achieve this, it is possible to reformulate the forward kinematics in a cylindrical coordinate of the end-effector, which is considerably more simple compared to the Cartesian representation:

$$\begin{bmatrix} r \\ \phi \\ z \end{bmatrix} = \begin{bmatrix} l_2 \cos q_2 + l_3 \sin q_3 \\ q_1 \\ l_2 \sin q_2 - l_3 \cos q_3 \end{bmatrix} \tag{6}$$

After this, we simply map the velocities:

$$\begin{bmatrix} u_x \\ u_\phi \end{bmatrix} = \begin{bmatrix} -C_x & 0 \\ 0 & -C_\phi \end{bmatrix} \begin{bmatrix} r \\ \phi \end{bmatrix} \tag{7}$$

E. Phantom Arm Control and Haptics

In order to control the arm, we can consider that the current position, $\vec{q}$, is the joint positions in which the human operator is holding the arm. The haptic feedback combines four factors:

1) Gravity & Viscosity Cancellation: To cancel the small-effects of gravity and viscous damping from the motor's back-EMF, we first implement a feed-forward term:

$$\vec{\tau}_c = G(\mathbf{q}) + \mathbf{K_R}\dot{\mathbf{q}} \tag{8}$$

This will negate the unwanted effects of gravity and damping, so that the human can purely feel the torques that we command.

2) Acceleration Feedback: We provide an negative feedback term from the vehicle's acceleration on top of this, such that

$$\mathbf{F}_a^p = -\mathbf{R}_b^p \mathbf{A} \mathbf{a}^b = -\mathbf{R}_b^p \begin{bmatrix} A_x & 0 & 0 \\ 0 & A_y & 0 \\ 0 & 0 & A_z \end{bmatrix} \begin{bmatrix} a_x^b \\ a_y^b \\ a_z^b \end{bmatrix} \tag{9}$$

This allows to simulate a fictitious inertial force, as if the user was actually driving a car. Here, $\mathbf{R}$ is always a fixed rotation that converts the car's body coordinates in Fig.2 to the Phantom's coordinate illustrated in Fig.3. This is described by the rotation matrix

$$\mathbf{R}_b^p = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{10}$$

3) Collision Detection: Also, we simulate a spring between the vehicle and an obstacle, if the vehicle gets too close to an obstacle. Given that the obstacle sensor returns a vector $\vec{d}$ of distance to the wall, all these sensors have a threshold distance $\vec{d_t}$ based on how far we want to warn the user there is going to be a collision. Assuming a unit vector $\hat{n}_i$ that points to the direction of this specific distance sensor, we choose the first distance sensor with the lowest value in

$\vec{d}$, such that

$$d_i = \min(\vec{d}) \tag{11}$$

After this, we simulate a physical spring attached to the vehicle.

$$\vec{F}_d = \begin{cases} -D(d_{ti} - d_i)\hat{n}_i & d_i \leq d_{ti} \\ 0 & d_i > d_{ti} \end{cases} \tag{12}$$

4) Zero-position Command: Finally, we simulate a "desired position" to be the zero-velocity position, in order to make the controller go to zero when there is no human input. This is done by a cartesian PD control,

$$\vec{F}_0 = -K_p(\vec{x} - \vec{x}_0) - K_d\dot{\vec{x}} \tag{13}$$

where the "zero" position is done by our home position of the joint angles,

$$\vec{x}_0 = f_{kin}(\vec{0}) \tag{14}$$

5) Final Torque Command: Combining all these factors, we linearly add the components to create our commanded torque:

$$\vec{\tau} = \vec{\tau}_c + \mathbf{J}^T(\vec{F}_a + \vec{F}_d + \vec{F}_0) \tag{15}$$

After the torque command is calculated, we sent it to each motor through utilizing the normalized PWM value

$$\vec{\alpha} = \frac{R}{V_{cc}}\frac{1}{K_T}\vec{\tau} \tag{16}$$

where R is the motor lead resistance, V_{cc} the voltage across the motor, and K_T the motor torque constant. With human input (human manually grabbing the phantom to a desired position), the torque commands do not actually affect the movement of the manipulator. In other words, the force that the human feels *is* our commanded torque. Thus the haptic force-feedback is

$$\vec{F} = \mathbf{J}^{-T}\left(\vec{\tau} - \vec{\tau}^c\right) = \vec{F}_a + \vec{F}_d + \vec{F}_0 \tag{17}$$

since our cancellation torque is no-longer felt by the human.

F. Without human: Stability

1) Controller Stability: To prove stability, we must prove that the vehicle converges to an end effector position $\vec{x}_0$ without an external force. Consider the following formulation where $u_x = v_x$ (or at least they linearly scale up by a constant factor). Then in the dynamics of the manipulator, we have

$$\mathbf{H}(q)\ddot{q} + \mathbf{C}(q,\dot{q})\dot{q} = \mathbf{J}^T(\vec{F}_a + \vec{F}_0) \tag{18}$$

On the acceleration term, we can see that the acceleration is connected to the joint position by

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \dot{v}_x - \omega_z v_y \\ \dot{v}_y + \omega_z v_x \\ 0 \end{bmatrix} = \begin{bmatrix} -C_x\dot{q}_3 \\ -C_x\frac{q_3^2}{L}\tan(-C_\phi q_1)) \\ 0 \end{bmatrix} \tag{19}$$

Therefore, the force feedback from the accelerometer is connected to the joint angles by

$$\mathbf{F}_a(q,\dot{q}) = -\mathbf{R}_b^p\mathbf{A}\vec{a}^b = \begin{bmatrix} -AC_x\frac{q_3^2}{L}\tan(-C_\phi q_1) \\ -AC_x\dot{q}_3 \\ 0 \end{bmatrix} \tag{20}$$

We can observe that the y component (mostly related to joint q_3 in the manipulator's coordinate frame) simply has a negative damping term. In contrast, we have a very interesting formula for the x component, which corresponds to sideways (mostly related to joint q_1) feedback. The theoretical behavior of this controller is illustrated through simulation in Section III.

Another option is a controller that will switch based on external input that can tell whether or not the human is grabbing the handle. This can be done through a pressure / force-sensitive resistor (FSR) sensor on the handle. Thus upon detecting that the human is no longer is grabbing the controller, the controller can switch to a simple PD control with desired position as home. Thus in this case we have

$$\vec{\tau} = \begin{cases} \vec{\tau}_c + \mathbf{J}^T(\vec{F}_a + \vec{F}_d) & R > R_{thres} \\ \vec{\tau}_c + \vec{F}_0 & R \leq R_{thres} \end{cases} \tag{21}$$

III. Simulation

A. Human Control: Accelerometer Feedback

First we simulate the joint trajectory of the phantom arm, effectively simulating human input to the arm. The designed joint trajectory is a sinusodial one, where the yaw joint goes through a full period (changing the steering direction), and the throttle joint accelerates the vehicle and comes to a stop. This joint trajectory is illustrated in Figure 4.

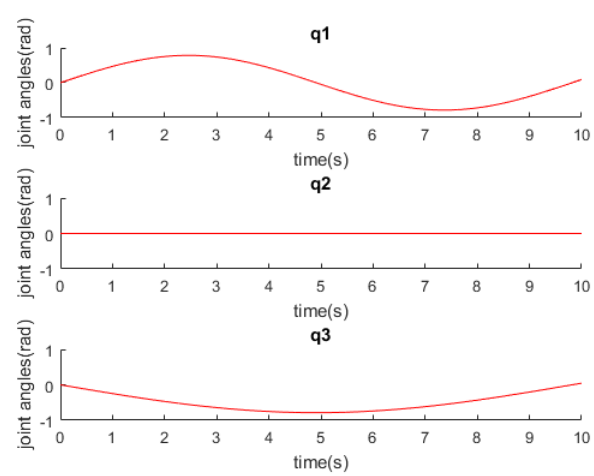

Fig. 4. Simulated Phantom Joint Trajectories by human input

Given the input in Figure 4, the vehicle path is illustrated in Figure 5, where the vehicle starts from zero position and zero angle.

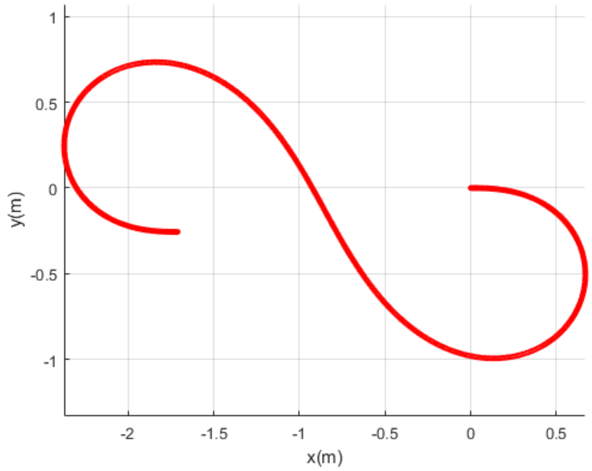

Fig. 5. Simulated Vehicle Trajectory given the Phantom Joint trajectory

We can see that the trajectory makes sense, given the the vehicle continuously goes forward, while changing the turning radius in the middle. After obtaining the vehicle velocities, we plot the theoretical acceleration feedback that the human feels in Figure 6.

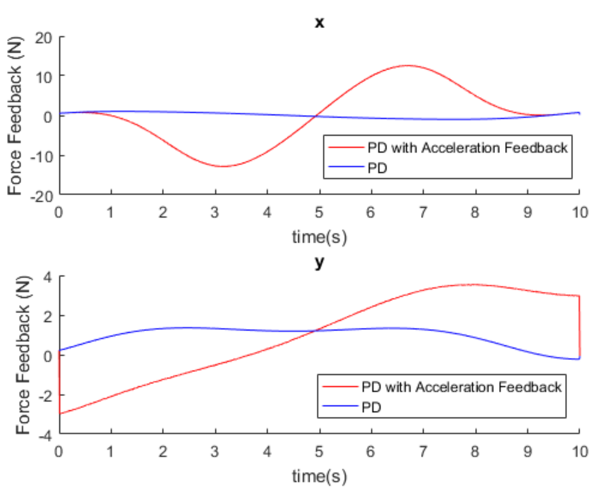

Fig. 6. Haptic Feedback to human given the Phantom Joint trajectory. The data is plotted for $K_p = 15, K_d = 5$, and $A = 20$

We can observe that from Figure 6 that the accelerometer term dominates over the PD term, which means the the human will feel major contribution from the acceleration of the vehicle instead of the PD term to desired home position. To illustrate this graph further, the x direction in Phantom's reference frame is related to the centrifugal acceleration that the human feels while turning in a circle. Fighter pilots who operate with similar joystick devices must also fight this inertial acceleration while they control the vehicle, and we can see that force feedback gives a more realistic input that simulates the vehicle's inertial forces in a non-inertial frame, as if the human was actually inside the vehicle.

B. Effect of Obstacles

In this section we illustrate the effect of obstacles, where the joint trajectory is simulated by the same trajectory in Figure 4 for q_3, while suppressing the effect of turning radius. At a certain time, the car gets close to an obstacle where the threshold distance is triggered, and a simulated spring kicks in to resist the motion of the car going forward. The force-feedback given this environment is illustrated in Figure 7.

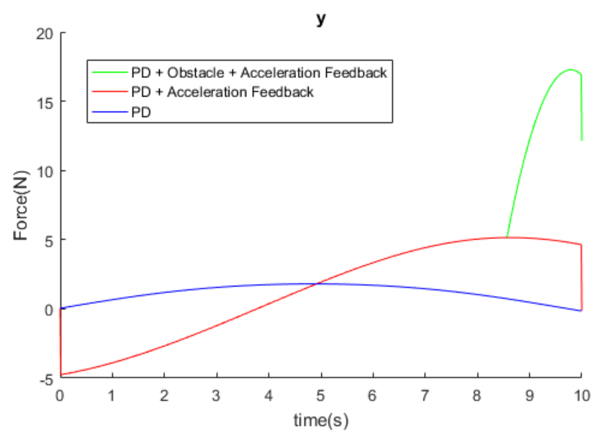

Fig. 7. Haptic Feedback to human given the Phantom Joint trajectory. The data is plotted for $K_p = 15, K_d = 5$, $A = 20$, and $D = 60$

At $t = 8.5$, the vehicle detects an obstacle in front of it, and a physical spring between the obstacle and the car is portrayed in the manipulator through cartesian force feedback. By assigning a higher gain to this spring, we impose a lot of penalty on obstacle collision.

C. Absence of Human Control: Controller Stability

Finally, we model the manipulator equation by the parameters for Phantom described in [6], and numerically solve this equation using fully implicit solvers. For a wide range of initial conditions and gains, we observed that the controller with accelerometer feedback does converge to the desired position of home, where it would command zero-velocity and stop the car. Figure 8 illustrates one of these cases with low damping gain.

We observe that while the joints to converge to our desired position of zero, the rate of convergence is significantly lower when accelerometer feedback is added. The difference this makes in terms of vehicle path is illustrated in Figure 9. It can be observed that with this configuration of gains, the vehicle travels $1.8m$ before it comes to a stop, while the simple PD control has the ability to stop the vehicle only within $\approx 0.2m$. To attempt to damp this further, we illustrate a different configuration of gains with a much higher gains in Figure 10.

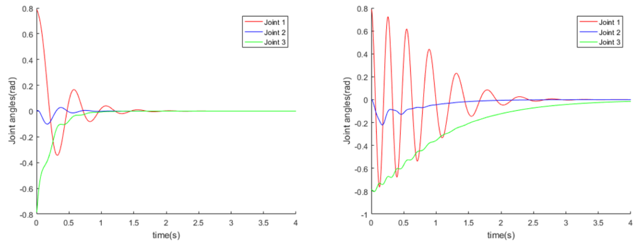

Fig. 8. A. PD control to home position with $K_p = 15$, $K_d = 0.5$. B. PD control with accelerometer feedback with additional $A = 20$

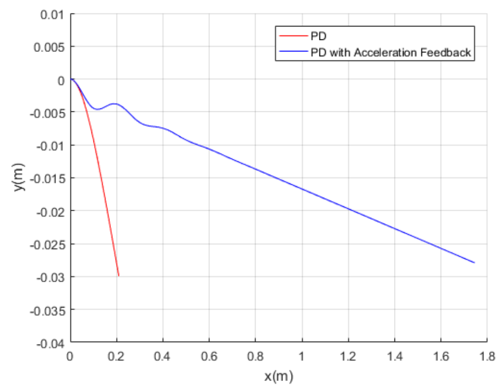

Fig. 9. Difference in vehicle path after the human lets go of the controller. Note that the x and y axis are not equal in scale, and the x component makes a major contribution.

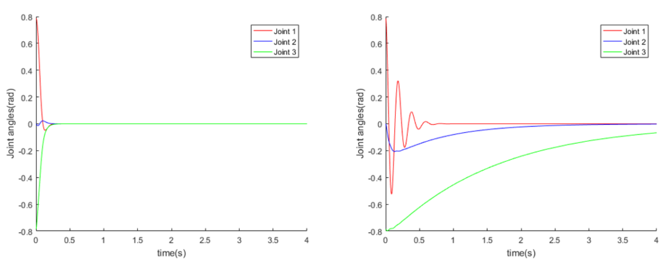

Fig. 10. A. PD control to home position with $K_p = 10$, $K_d = 5$. B. PD control with accelerometer feedback with additional $A = 20$

We can also observe that despite increased damping provided by the PD controller, the relative rate of convergence shows no significant change, and shows similar behavior. Finally, we illustrate the effect of gain A in Figure 11 by lowering the accelerometer gain and observing the change in response time. Under the same PD gains, a comparison of Figure 10.B and Figure 11 clearly illustrates that the accelerometer feedback has a significant impact on the rate of convergence, and lower accelerometer lead to faster rates of convergence.

Through these simulations we conclude that in order to achieve higher rate of convergence for our safety criteria, it is desirable to have a relatively low value of A. However, there exists a tradeoff between K_p, since with a low value of A, we

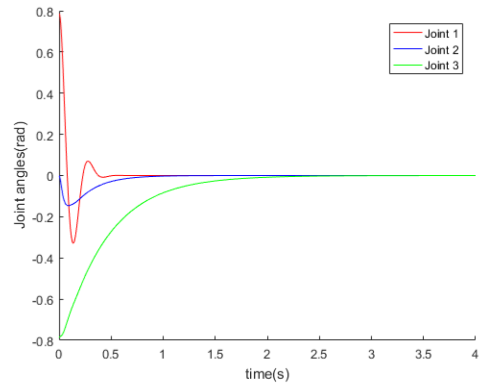

Fig. 11. PD Control with accelerometer feedback with $K_p = 100$, $K_d = 5$, $A = 5$

cannot arbitrarily impose a high value of K_p (otherwise, the PD control term will be much higher than the accelerometer term in force feedback, and provide a less interesting force feedback to the human operating the device). This means how strong we decide to make the gains for this controller is fundamentally limited by the accelerometer feedback term, and its rate of convergence.

We finally remark that observing Figure 9, because pure PD control has much better rate of convergence and thus much less travel distance in vehicles, it is desirable to purely use the PD controller when the human accidentally lets go of the controller. Therefore implementing another method of sensing the human grip (other automatic sensing or manual human input through buttons) is more desirable.

IV. Hardware Setup & Experiment

In order to carry out a demonstration, we mounted an electronics platform on top of the Traxxas X-Maxx vehicle, and hijacked the ESC signals in order to command the vehicle. The resulting vehicle contains a NVidia TX-2 Processor, an Phidgets Inertial-Measurement Unit (IMU), a hobby-level lidar (RPLidar A20), multiple ultrasonic sensors for collision detection, and a RGB-D camera for perception and visual odometry. The vehicle is illustrated in Fig.12.

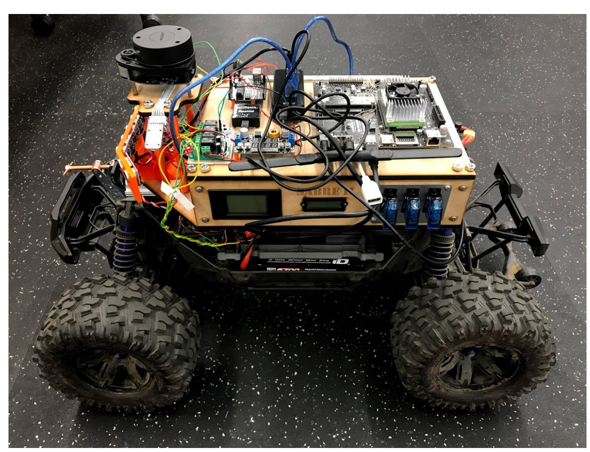

Fig. 12. Experimental Vehicle Platform

In order to change the phantom manipulator into a tele-operator system, we mounted it upside down with a handle, and relayed the joint information through a ground-station (laptop), and communicated to the TX-2 through the wifi. The Phantom platform is shown in Fig.13, controlled by the Teensy microcontroller, and the Roboclaw motor controller.

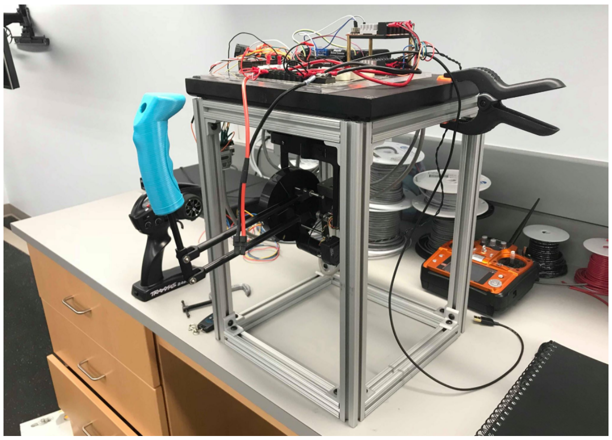

Fig. 13. Experimental Vehicle Platform

With the tuned gains relating the accelerometer to the phantom manipulator, the vehicle showed relatively stable behavior even without human intervention. However, for safety reasons, the throttle was commanded externally by a RC controller, while the phantom only controlled the steering angle of the vehicle. The negative feedback from the accelerometer proved to be quite an intuitive feedback that relayed the vehicle's inertial state. For instance, it was possible to feel obstacles (bumps), slopes, and centrifugal force induced due to turning. Finally, the collision-feedback from the ultrasonic sensors proved to be an interesting form of warning to the user as well.

V. References

References

[1] M. Yamashita, "Assistive Driving Simulator with Haptic Manipulator Using Model Predictive Control and Admittance Control", 6th International Conference on Intelligent Human Computer Interaction (IHCI) 2014

[2] N. Diolati, C. Melchiorri, "Tele-operation of mobile robot through haptic feedback", IEEE International Workshop on Haptic virtual Environments and Their Applications", 2002

[3] D. Kim, K.W. Oh, D. Hong, Y.K. Kim, S.H. Hong, "Motion Control of Excavator with Tele-Operated System", 26th International Symposium on Automation and Robotics in Construction (ISARC), 2009

[4] R. Murray, Z. Li, S. Sastry, "A Mathematical Introduction to Robotic Manipulation",

[5] M.C. Cavusoglu, D. Feygin, "Kinematics and Dynamics of Phantom (TM) Model 1.5 Haptic Interface", Sensable, pp.1-13

VI. Appendix

A. Forward Kinematics and Jacobian

For the Phantom Manipulator, the forward kinematics can be obtained as a function of three angles $[\theta_1, \theta_2, \theta_3]$:

$$\mathbf{x} = \begin{bmatrix} -\sin\theta_1(a\cos\theta_2 + b\sin\theta_3) \\ \cos\theta_1(a\cos\theta_2 + b\sin\theta_3) \\ a\sin\theta_2 - b\cos\theta_3 \end{bmatrix} \tag{22}$$

Of which the Jacobian that relates the joint velocities to the end-effector velocity is given by

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\theta} \qquad \mathbf{J} = \frac{\partial \mathbf{x}}{\partial \theta} \tag{23}$$

which in full form is given by

$$\begin{bmatrix} -\cos\theta_1(a\cos\theta_2 + b\sin\theta_3) & a\sin\theta_1\sin\theta_2 & -b\sin\theta_1\cos\theta_3 \\ -\sin\theta_1(a\cos\theta_2 + b\sin\theta_3) & -a\cos\theta_1\sin\theta_2 & b\cos\theta_1\cos\theta_3 \\ 0 & a\cos\theta_2 & b\sin\theta_3 \end{bmatrix} \tag{24}$$