2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)
November 3 - 5, 2015, Seoul, Korea

Achievement of Recognition Guided Teleoperation Driving System for Humanoid Robots with Vehicle Path Estimation

Iori Kumagai, Ryo Terasawa, Shintaro Noda, Ryohei Ueda, Shunichi Nozawa, Yohei Kakiuchi, Kei Okada, Masayuki Inaba
Graduate School of Information Science and Technology, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, 113-8656 Tokyo, Japan
Email: iori, terasawa, s-noda, ueda, nozawa, youhei, k-okada, inaba@jsk.t.u-tokyo.ac.jp

***Abstract*—In the wake of the DARPA Robotics Challenge, the task for robots to drive vehicles has been expected to be a method for robots to transport themselves to the disaster site where it is hazardous for humans to approach. In the driving task, it is important for the robot to estimate the path of the vehicle and select an appropriate path for navigation through unknown obstacles, even under limited communication with an operator. It is also necessary for robots to suggest the estimated path of vehicle to an operator to deal with unforeseen circumstances. Therefore, we propose a recognition guided teleoperated driving system for robots to drive vehicles in disaster sites with estimated vehicle path based on steering angle and vehicle model. First, we show model based steering and pedaling strategy to achieve the target steering angle for desired path. Next, we propose a vehicle path estimation and a local planner that can suggest a traveling path according to the surroundings. We integrated them into a teleoperation system for bandwidth limited environments as recognition guidance. Finally, we show the effectiveness of our driving system by conducting field driving experiments with three different robots: JAXON, STARO and HRP-2.**

I. Introduction

Fig. 1. JAXON [1] on a Polaris RANGER XP900 [2]

In the wake of DARPA Robotics Challenge [3], the task of robots to drive vehicles has been attracting attention. Robots to drive a vehicle is a challenging task which includes problems such as tool manipulation like steering or pedaling, motion planning in narrow driver's seat for getting on and off, sitting posture control with whole body environmental contacts, recognition of obstacles, self-localization and path planning for driving, system integration for automation and teleoperation.

Especially, driving task in disaster response support is expected to be a method for robots to transport themselves to the disaster site where it is hazardous for humans to approach. In a driving task for disaster response support, it is important that the robot can estimate the vehicle path and receive instructions from an operator if necessary, because there may be some problems such as poor communication and narrow paths caused by unknown obstacles which can be corrected online by the operator.

In this paper, we propose the configuration method for a driving system which includes teleoperation and autonomous path planning using vehicle path estimation and discuss about the use of the system for robots to drive vehicles in the disaster site as shown in Fig.1.

II. Requirements and Strategies of Driving System for Humanoid Robots

Driving vehicles is a complex task for humanoid robots which includes recognition, control and system design. However, past research studies mainly focused on the individual elements separately. For example, autonomous driving with recognition has been studied for a long time. Knowledge on recognition strategy for autonomous vehicles such as obstacle detection and path planning is accumulated in DARPA Urban Challenge [4], [5]. Furthermore, motion planning in narrow spaces such as driver's seat by Yokokohji et al. [6], whole body manipulation for crank operation by Nishiwaki et al. [7] and teleoperation system for heavy machines by Yokoi et al. [8] are examples of related works for fundamental technologies. Integration systems of autonomy and teleoperation have also been discussed long time for the level of autonomy of the system. Some methodologies were suggested such as shared autonomy [9] and supervised autonomy [10]. In disaster sites, it is desirable to be able to adjust the level of autonomy depending on communication latency and field conditions.

On the other hand, research on vehicle driving by humanoid robots as an integrated complex task become active

978-1-4799-6885-5/15/$31.00 ©2015 IEEE

recently. Rasmussen et al. enabled humanoid robots to drive vehicle by a control system that includes steering using a peg-in-wheel method, pedaling motion using ankle pitch joint and path planning with cost map [11]. However, peg-in-wheel steering motion includes backlash and it may cause difference of traveling path between a model in the controller and a real vehicle. If the operator sends erroneous instruction referencing wrong steering information, a robot may crash into obstacles.

Paolillo et al. dealt with vehicle driving task as an integration problem of steering manipulation and recognition on open road [12]. They detect boundary lines of a road using edge detection and achieved autonomous driving with compliance control in simulation environment. They proposed a method for determining target steering angle using vanishing point of boundary lines and integrated it into steering manipulation of a robot, although, they do not consider obstacles on the open road.

In disaster sites, it is assumed that quality of communication is poor due to latency and packet loss, and that there are many unknown obstacles creating narrow paths. Although Kitt et al. proposed a method to estimate linear and angular velocities of the vehicle from a sequence of stereo images and calculate its odometory [13], there is no guarantee that the vehicle can travel enough distance to collect enough samples for odometory in disaster sites which have a lot of obstacles. Roberts et al. proposed the obstacle detection method using superpixel labeling with optical flow templates [14]. However, the robot must not only detect obstacles but also estimate the traveling path of the vehicle and select an appropriate path in order to drive a vehicle by itself. Furthermore, it is also important in disaster response scenarios that a driver robot can suggest the estimated path of the vehicle to the operator in order to obtain appropriate instructions for dealing with unforeseen circumstances such as change of destination due to unexpected obstacles.

Based on these literatures, we propose a driving system that combines the use of autonomy and teleoperation as shown on the left of Fig.2 for a robot to drive a vehicle in disaster sites. In the following sections, we first present a model based steering strategy for achieving target steering angle and a pedaling strategy for achieving the ideal traveling path. Next, we propose a vehicle path estimation method using the steering angle of vehicle and a reactive local planner for suggesting an ideal path to avoid obstacles based on instantaneous point clouds from the camera of the robot. We call these vehicle path information from the robot as recognition guidance and integrate them into teleoperation system in bandwidth limited environments, which imitates communication in disaster sites. In our driving system, an operator can change the priority of recognition guidance by toggling between steering command from the operator and the local planner. Finally, we verify our driving system by conducting field driving experiments with three different robots and prove its usefulness.

III. Equipment and Control Strategy for Driving

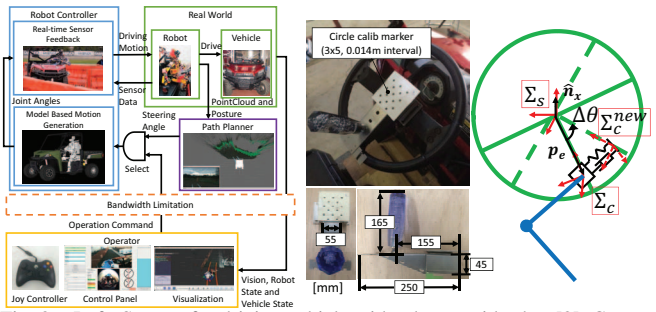

Fig. 2. Left: System for driving vehicle with a humanoid robot [2], Center: The crank equipment for steering, Right: Steering with impedance control and angle velocity limitation

A. Steering Strategy

In vehicle driving tasks, steering angle determines traveling direction of the vehicle. Therefore, it is important in vehicle driving task that the driver knows the current steering angle and is able to turn steering wheel to achieve the target steering angle. It is also necessary that the driver is able to steer quickly while driving a vehicle. There are previous research studies that generate turning motions for dual-arm robots using a grasp-rotate-release-regrasp method [15]. However we made a crank attachment which is shown in the center of Fig.2 to achieve both high turning speed and wide operation range.

The crank attachment has a calibration marker board, and the robot calculates the relative coordinates of the steering center Σ_s by detecting the marker board after sitting in the vehicle. The coordinates of the crank handle Σ_c can be calculated from Σ_s, and the robot grasps the crank handle using Σ_c.

Since it is not possible to get the current actual steering angle without using a special tool such as an OBD (On-Board Diagnostics) for regular vehicles, the current steering angle is estimated using the steering center coordinates Σ_s and the current hand coordinates of the robot Σ_e. Assuming that the robot is grasping the crank handle, the hand position in the steering center coordinate on the steering plane of the wheel can be used to estimate the current steering angle. In other words, the current steering angle is calculated according to Eq.1 using the relative hand position from the wheel center position $\boldsymbol{p}_{\boldsymbol{e}}^{\boldsymbol{\Sigma}_s}$ and the unit vector $\hat{\boldsymbol{n}}_{\boldsymbol{x}}^{\boldsymbol{\Sigma}_s}$ in the x-axis of Σ_s.

$$\theta = \arccos\left(\frac{\boldsymbol{p}_{\boldsymbol{e}}^{\boldsymbol{\Sigma}_s} \cdot \hat{\boldsymbol{n}}_{\boldsymbol{x}}^{\boldsymbol{\Sigma}_s}}{\|\boldsymbol{p}_{\boldsymbol{e}}^{\boldsymbol{\Sigma}_s}\|}\right) - \theta_0 + 2\pi n \qquad (1)$$

In Eq.1, θ_0 is the angle formed by $\boldsymbol{p}_{\boldsymbol{e}}^{\boldsymbol{\Sigma}_s}$ and $\boldsymbol{n}_{\boldsymbol{x}}^{\boldsymbol{\Sigma}_s}$ when the steering angle is zero and n is determined as the integer value that minimizes the difference between θ and the estimated steering angle one control cycle before.

Steering motion is generated using a model based approach. The target steering angle θ_d is given selectively by path planner, which is explained in Section IV, or teleoperation command, which is explained in Section V. The robot calculates the rotation angle of steering $\Delta\theta$ for the next

control cycle after Δt according to Eq.2 using θ_d and the current steering angle θ with angular velocity limit ω_{max}.

$$\Delta\theta = \mathrm{sgn}(\theta_d - \theta) min(\omega_{max}\Delta t, abs(\theta_d - \theta)) \quad (2)$$

Using $\Delta\theta$, we calculate the reference coordinates of the robot's hand Σ_c^{new}, and set that as reference coordinates of the impedance controlled arm [16] in real-time controller. After the arm settles to the new position Σ_c^{target}, to relieve any internal forces and moments caused by errors between the model and real robot, we measure the resulting joint angles of the robot considering self collision and set it as the command joint angles. The robot achieves the target steering angle by generating steering motion which is explained above iteratively.

B. Pedaling Strategy

The method of generating pedaling motion is dependent on the hardware configuration of the robot. The robots which have long legs like JAXON [1] and STARO [17] use only ankle pitch joint for pedaling to minumize the number of joints involved since they can reach the gas pedal with their long legs but are more prone to collision with the vehicle when moving in the narrow driver's seat. On the other hand, the robots which have short legs like HRP-2 [18] use a pedaling pedestal to reach the gas pedal. We give a pedaling command p, where $0.0 \leq p \leq 1.0$, and define a mapping function χ for each robot which maps the pedaling command to joint angles of the robot leg Θ_p. In this paper, we define the maximum amount of depression as 1.0 and minimum amount of depression as 0.0 in terms of ankle pitch joint for JAXON and foot height of right leg for HRP-2, and interpolate linearly between maximum and minimum values.

IV. Recognition Guidance for Driving

In teleoperation, it is difficult for the operator to indicate an appropriate path only with images from a robot since the operator would have to estimate how the vehicle moves by himself. Therefore, we estimate the path of the vehicle based on the steering angle and the vehicle model as guidance for the operator. Furthermore, we propose a local planner that can suggest a target steering angle for an appropriate path as more automated guidance.

A. Vehicle Path Estimation

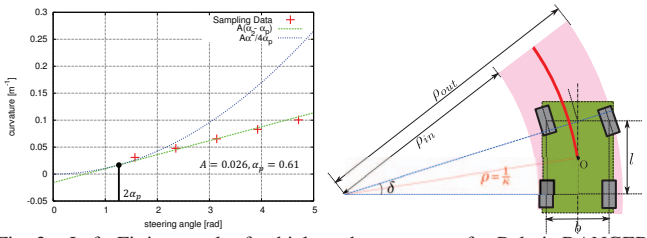

Fig. 3. Left: Fitting result of vehicle path parameters for Polaris RANGER XP900, Right: Ackerman steering geometry

The radius of the estimated vehicle path is calculated from the relationship between current steering angle α and turning curvature κ. We estimated the transfer function $\kappa = f(\alpha)$ by sampling α and measuring κ in steady-state circular turning using a real vehicle. The result of sampling and fitting for Polaris RANGER XP900 are shown on the left of Fig.3. The relationship between α and κ of Polaris RANGER XP900 is almost linear in the range of $\|\alpha\| \geq \alpha_p$ but there is a dead zone in $\|\alpha\| \leq \alpha_p$ where the vehicle does not turn. In order to preserve continuity and smoothness of f, we define f as Eq.3. α_p and A can be defined by least-squares fitting in $\|\alpha\| \geq \alpha_p$ as the left of Fig.3. Still turning curvature in steady-state circular turning depends on steer characteristic and speed of vehicle. We assumed that speed of the vehicle is slow enough such that influence of slippage caused by centrifugal force is negligible.

$$\kappa = f(\alpha) = \begin{cases} A(\alpha - \alpha_p) & \alpha \geq 2\alpha_p \\ \mathrm{sgn}(\alpha)\frac{A}{4\alpha_p}\alpha^2 & \|\alpha\| \leq 2\alpha_p \\ A(\alpha + \alpha_p) & \alpha \leq -2\alpha_p \end{cases} \quad (3)$$

The vehicle coordinates relative to the robot are also needed to display the estimated vehicle path. We attached a marker board on the center of the vehicle dashboard and calculated the relative coordinates from the robot to the vehicle by detecting the marker board.

B. Obstacle Distance on Estimated Vehicle Path

We can estimate the direction to the nearest obstacle on the estimated path using the point cloud data. First, we obtain the 3D point cloud $\mathcal{C}_{all}^{camera}$ from the head camera of the robot and transform it to the coordinate which is fixed to the center of the vehicle. Second, we extract obstacle point cloud $\mathcal{C}_{obstacle}^{car\_center}$ by removing point cloud points that are within the bounding boxes of vehicle or the ground. The bounding boxes of the vehicle and the ground are defined in prior. Next, we extract the point cloud points $\mathcal{C}_{obstacle_\kappa}^{car\_center}$ that lies on the estimated vehicle path by performing range search using kd-tree algorithm. The center of turning path is calculated as $\boldsymbol{c} = \left(-\frac{l}{2}, \sqrt{\rho^2 - (\frac{l}{2})^2}, 0\right)$ and radii of outside path ρ_{out} and inside path ρ_{in} are calculated as Eq.4 from the Fig.3. $\Delta\rho_{in}$ and $\Delta\rho_{out}$ are the inside and outside turning radius differences, and b and l are the width and length of a vehicle respectively. The desired point cloud $\mathcal{C}_{obstacle_\kappa}^{car\_center}$ is bounded by the circle of radius ρ_{out} and the circle of radius ρ_{in} centered on $\boldsymbol{c}$.

$$\rho_{in} = \rho - \frac{b}{2} - \Delta\rho_{in}, \ \rho_{out} = \rho + \frac{b}{2} + \Delta\rho_{out} \quad (4)$$

Finally, we calculate the nearest obstacle point from the center of the vehcile in $\mathcal{C}_{obstacle_\kappa}^{car\_center}$ using nearest neighbor search in a kd-tree, and we obtain the distance d from the center of the vehicle to the obstacle along the path. If there are no obstacles along the path, we set d to be the maximum range of the camera.

C. Adaptive Path Suggestion by Local Planning

In this section, we propose a driving focused method for local planning based on Dynamic Window Approach(DWA) [19]. DWA is a local planning method that calculates the optimal linear and angular velocities for the next control cycle using obstacle position and the distination. Our local planner takes only current obstacle position and distination and does not use global information for disaster environment, where it is difficult to obtain prior information, and outputs target steering angle.

We first generate N paths (where N is an odd number, with one straight path, and equal number of paths to the left and right shown as Fig.4), between a pre-defined max and min steering angles $\alpha_{max}, \alpha_{min}$. We then evaluate each path according to a scoring function based on three factors: Obstacle Factor, Heading Factor and Difference Factor.

As background knowledge, we define the relationships between steering angle α, which is the angle of steering wheel for driver, and actual steer angle δ, which is the angle between the front of the vehicle and the direction of wheels. The mapping function $\delta = g(\kappa)$ depends on the speed of the vehicle and its steer characteristics. However, we define g as $\delta = g(\kappa) = l\kappa$ using the geometic relation as shown on the right of Fig.3 assuming that speed of the vehicle is slow enough such that influence of slippage caused by centrifugal force can be ignored [20]. In this equation, we adopt a linear approximation because δ is small enough in the practical range of steering angle according to the measurement results as shown in the left of Fig.3.

Using mapping function f from steering angle α to turning curvature κ defined in IV-A and g, we can define α, δ, κ when one of three value is given.

1) Mapping from Path to Real Steer Angle: The real steer angle δ_n for path n is calculated as Eq.5 using $\delta_{max} = g\left(f(\alpha_{max})\right)$ and $\delta_{min} = g\left(f(\alpha_{min})\right)$. Then, κ_n and α_n can be calculated using δ_n, f^{-1} and g^{-1}.

$$\delta_n = \delta_{max} - \frac{\delta_{max} - \delta_{min}}{N-1}(n-1) \tag{5}$$

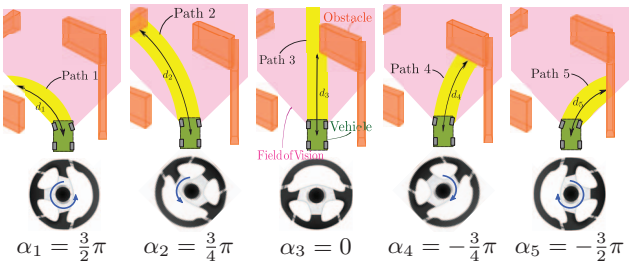

Fig. 4. Relation between steering wheel angle and path, and distance to the obstacles d_n. This is the example of $\alpha_{max} = \frac{3}{2}\pi, \alpha_{min} = -\frac{3}{2}\pi, N = 5$

2) Obstacle Factor: The path which has the longest distance to the nearest obstacle is the most appropriate for driving. Therefore, we calculate distance to the nearest obstacle d_n for each path by the method shown in Subsection IV-B and normalize them to the range of 0.0 to 1.0. We used the normalized distance to define an obstacle factor (OF). OF is calculated as Eq.6, where d_{min} is the minimum distance and d_{max} is the maximum distance in all d_n.

$$\mathrm{OF}_n = \sqrt{\frac{d_n - d_{min}}{d_{max} - d_{min}}} \tag{6}$$

3) Heading Factor: The obstacle factor contributes to avoiding obstacles but it is not guaranteed that it is in the direction of the goal. Therefore, we consider the goal direction using a heading factor (HF). We calculate HF_n as Eq.7 using the goal direction φ in the coordinate in the center of the vehicle and δ_n.

$$\mathrm{HF}_n = 1 - \frac{\|\delta_n - \varphi\|}{2\pi} \tag{7}$$

4) Difference Factor: Angular velocity of steering $\dot{\alpha}$ is limited by joint velocity limitations of the driver robot. Therefore, we calculate a difference factor DF_n as Eq.8 to prevent target steering angle from jumping.

$$\mathrm{DF}_n = 1 - \frac{\|\alpha - \alpha_n\|}{\alpha_{max} - \alpha_{min}} \tag{8}$$

5) Selection of Appropriate Path: We define the score $J(n)$ of each path n as Eq.9 and select the path n_{max} that has the maximum score. ξ, η and ζ are weight parameters for our local planner. The output of our local planner is $\alpha_{n_{max}}$ which is steering angle of path n_{max} and this output is used as the target steering angle θ_d in the steering motion if the local planner suggestion is selected.

$$J(n) = \xi \cdot \mathrm{OF}_n + \eta \cdot \mathrm{HF}_n + \zeta \cdot \mathrm{DF}_n \tag{9}$$

V. Communication and Instruction Interfaces in Teleoperation

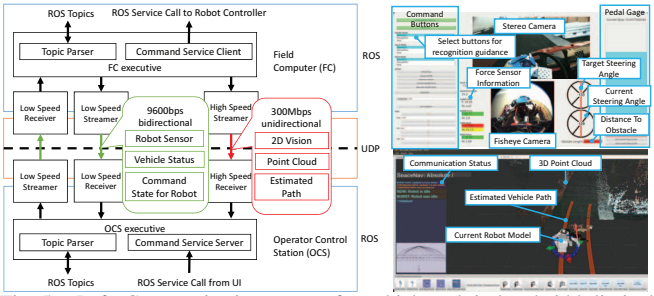

Fig. 5. Left: Communication system for vehicle task in bandwidth limited environment. This shows communication cannels between robots, field computers (FC) and an operator control station (OCS), Right: User interface for vehicle task (Upper: Control panel, Lower: Visualization interface)

A. Communication in a bandwidth limited environment

In the vehicle driving task of the DARPA Robotics Challenge, 300Mbps and 9600bps channels were provided for communication from the robot to the operators, and only a 9600bps channel was provided for communication from operators to robots. Therefore, we built a communication system shown in the left of Fig.5. We used ROS system independently in field computers (FC) and operator control stations (OCS) [21]. All data between FC and OCS are summarized into one ROS topic in each channel by "executive" and encoded into UDP packets by "streamer". The

UDP packets are received and decoded into ROS messages by "receiver" and parsed and published by "executive". Large data such as point clouds and 2D images are sent through the 300Mbps channel, and small or bidirectional data such as sensor information, status of the vehicle and status of operation commands are sent through the 9600bps channel. The execution status (such as starting, executing and finishing) of motion commands sent from the control panel is checked by the command service server and client in "executives" to monitor the progress of the motion.

Ideal update cycle of images and estimated vehicle path are 10Hz, command and status of vehicle are 5Hz and robot status is 1Hz. However, there may be large latency in communication under limited bandwidth and the delayed images caused by this latency can result in the operator making incorrect decisions. A elapsed time from recent update of each channel is displayed as shown in the lower right of Fig.5 to prevent aformentioned problem.

B. Control Panel and Visualization for Teleoperation

We used a control panel and visualization interface for teleoperation as shown in the right of Fig.5. An operator sends commands through buttons in the control panel and a joystick controller referencing point clouds and estimated vehicle path in the visualization interface. Estimated vehicle path is overlayed onto point clouds and camera images from the robot to tell the operator how the vehicle will move with the current steering angle. The operator can instruct an appropriate path easily to avoid obstacles referencing the overlayed estimated vehicle path.

To supress the traffic in the presence of bandwidth limitation, the series of motions such as grasp/release crank handle can be commanded from buttons in the control panel. The sensor information from the robot and state of vehicle such as steering angle, amount of pedaling and distance to the nearest obstacle are visualized in the control panel. An operator sends target steering angle θ_d and command for amount of pedaling p through the joystick controller. The operator also can select to use the local planner suggestion explained in Subsection IV-C or operator indication as target steering angle from command buttons.

In vehicle driving tasks, it is necessary for the operator to see the side of the vehicle in order to avoid collisions between the vehicle and obstacles, but it is difficult to detect obstacles on the sides of the vehicle using only the head camera of the robot. Therefore, we attached a fish-eye lens camera on the robot's chest to detect the side obstacles.

VI. Achievement of Driving Tasks in Real World

A. Vehicle Driving using Teleoperation

We carried out a driving experiment with JAXON and Polaris RANGER XP900 using the proposed driving system. The result is shown in Fig.6. A operator sends commands using only the control panel and visualization interfaces shown on the left of Fig.6 from a remote operator station. In this experiment, we impose the same bandwidth limitations as those used in the DRC Finals between the robot and the operator station, and rely mainly on having an operator commands using the overlayed estimated vehicle path on the point cloud data. The operator used only teleoperation and JAXON succeded in driving the vehicle around two obstacles in approximately 120[sec].

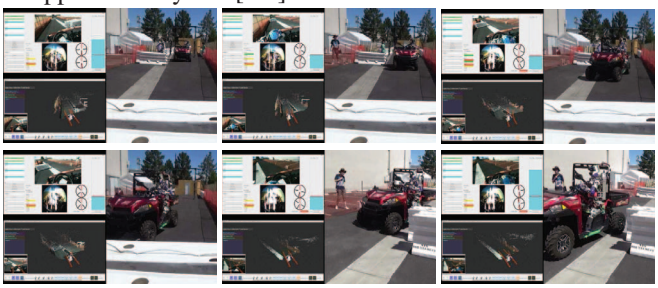

Fig. 6. JAXON drives a vehicle by teleoperation. Left shows user interface for operator and right shows driving robot in real world

B. Vehicle Driving using Autonomous Path Planning

We carried out a driving experiment with STARO and Polaris RANGER XP900 using the automatic steering angle determination method with the proposed local planner. The parameters for the local planner were $\xi = 10.0, \eta = 7.5, \zeta = 4.0$ and the course has six turns with a total length 90[m]. In this experiment, we used the proposed local planner to determine target steering angle putting high priority on the autonomous recognition guidance and sent commands only for pedaling and looking around to detect obstacles.

The result is shown in Fig.7. The robot completed the task in approximately 16[min] without any instructions on the steering angle from the operator.

Fig. 7. Snapshots of driving experiment with the humanoid robot STARO and the Polaris RANGER XP900. Upper left shows suggested path as blue lines and obstacle point clouds as green points, bottom left shows the camera image of STARO and right shows driving robot in real world.

C. Vehicle Driving Task in the DRC Finals

In DRC Final, JAXON and HRP-2 attempted the vehicle driving task on the Polaris RANGER XP900 relying fully on tele-operation [22]. Both robots completed the vehicle driving task successfully with JAXON taking 140[sec] and HRP-2 taking 310[sec]. Fig.8 shows the snapshots of JAXON performing the vehicle driving task and Fig.9 shows those of HRP2 performing the vehicle driving task.

VII. Conclusion

In this paper, we proposed a practical teleoperation driving system with autonomous recognition guidance. We achieved

Fig. 8. JAXON drives a vehicle in DARPA Robotics Challenge [23]

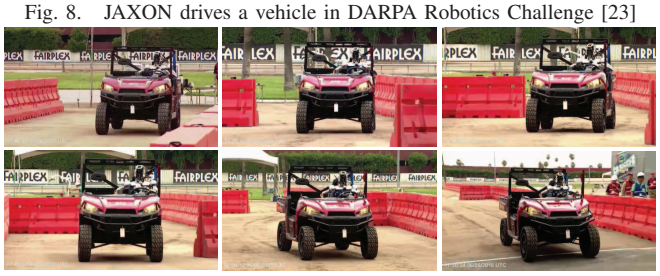

Fig. 9. HRP-2 drives a vehicle in DARPA Robotics Challenge [23]

target steering angle and high steering speed using a crank attachment and proposed pedaling methods suitable for different robots. To enable a vehicle to move in an appropriate path, we proposed a vehicle path estimation method and a simple local planner for determining target steering angle. We integrated them to a teleoperation system for disaster sites as recognition guidance and verified the usefulness by completing vehicle driving tasks with three different robots.

Our proposed driving system enables a robot to automously suggest the appropriate path while still allowing an operator to indicate an appropriate target steering angle. In this paper, we showed the proposed method can be applied to disaster sites with poor communication situation and a lot of obstacles. This system also gives the operator room to modify autonomy level by selecting steering command depending on communication latency and field condition.

As future work, we need to develop a self determination method for the robot to assign priorities to teleoperation and autonomous control in order to increase task feasibility in the situation where the operator cannot indicate commands.

ACKNOWLEDGEMENT

The achievement in this paper is partially gained as a result of the commissioned project of New Energy and Industrial Technology Development Organization (NEDO) “Research and Development of Biped Dual-Armed Humanoid Robot System for Disaster Response”.

REFERENCES

[1] Kunio Kojima, Tatsuhi Karasawa, Toyotaka Kozuki, Eisoku Kuroiwa, Sou Yukizaki, Satoshi Iwaishi, Tatsuya Ishikawa, Ryo Koyama, Shintaro Noda, Fumihito Sugai, Shunichi Nozawa, Yohei Kakiuchi, Kei Okada, and Masayuki Inaba. Development of life-sized high-power humanoid robot jaxon for real-world use. In *Proceedings of the 2015 IEEE-RAS International Conference on Humanoid Robots*.

[2] DRC Garally. http://www.theroboticschallenge.org/gallery-all.

[3] DRC Finals. http://www.theroboticschallenge.org/.

[4] Jörn. W. M and Thomas F. Realizing complex autonomous driving maneuvers the approach taken by team CarOLO at the DARPA urban challenge. In *Proceedings of the 2008 IEEE International Conference on Vehicular Electronics and Safety*, pages 232–236, 2008.

[5] Yoshiaki Kuwata, Justin Teo, Gaston Fiore, Sertac Karaman, Emilio Frazzoli, and Jonathan P. How. Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, 17(5):1105–1118, 2009.

[6] Yasuyoshi YOKOKOHJI, Shigemitsu NOMOTO, and Tsuneo YOSHIKAWA. Static evaluation of humanoid robot postures constrained to the surrounding environment through their limbs. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2:1856–1863, 2002.

[7] K. Nishiwaki, S. Kagami, and H. Inoue. Object manipulation by hand using whole-body motion coordination. *IEEE International Conference Mechatronics and Automation*, 4:1778–1783, 2005.

[8] K Yokoi, K Nakashima, M Kobayashi, H Mihune, H Hasunuma, Y Yanagihara, T Ueno, T Gokyuu, and K. Endou. A tele-operated humanoid robot drives a backhoe in the open air. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2:1117–1122, 2003.

[9] Paul Michelman and P. Allen. Shared autonomy in a robot hand teleoperation system. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, 1994.

[10] Gordon Cheng and Alexander Zelinsky. Supervised autonomy: a framework for human robot systems development. *Information Sciences*, 10(3):251–266, 2001.

[11] Christopher Rasmussen, Kiwon Sohn, Qiaosong Wang, and Paul Oh. Perception and control strategies for driving utility vehicles with a humanoid robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 973–980. Ieee, September 2014.

[12] Antonio Paolillo, Andrea Cherubini, Abderrahmane Kheddar, and Marilena Vendittelli. Toward Autonomous Car Driving by a Humanoid Robot : A Sensor-Based Framework. In *IEEE/RAS International Conference on Humanoid Robots*, pages 451–456, 2014.

[13] Bernd Kitt, Andreas Geiger, and Henning Lategahn. Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme. *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 486–492, 2010.

[14] Richard Roberts and Frank Dellaert. Direct Superpixel Labeling for Mobile Robot Navigation Using Learned General Optical Flow Templates. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1032–1037, 2014.

[15] Arash Ajoudani, Jinoh Lee, Alessio Rocchi, Mirko Ferrati, Enrico Mingo Hoffman, Alessandro Settimi, Darwin G Caldwell, Antonio Bicchi, and Nikos G Tsagarakis. A Manipulation Framework for Compliant Humanoid COMAN : Application to a Valve Turning Task. In *IEEE/RAS International Conference on Humanoid Robots*, pages 664–670, 2014.

[16] Neville Hogan. Impedance control: An approach to manipulation. In *American Control Conference*, volume 107, pages 304–313, 1984.

[17] Yoshito Ito, Shunich Nozawa, Junichi Urata, Takuya Nakaoka, and Kazuya Kobayashi. Development and Verification of Life-Size Humanoid with High-Output Actuation System. In *IEEE International Conference on Robotics and Automation*, pages 3433–3438, 2014.

[18] Kenji Kaneko, Fumio Kanehiro, Shuuji Kajita, Hirohisa Hirukawa, Toshikazu Kawasaki, Masaru Hirata, Kazuhiko Akachi, and Takakatsu Isozumi. Humanoid robot HRP-2. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2:1083 – 1090, 2004.

[19] D. Fox, W. Burgard, and S. Thrun. Controlling synchro-drive robots with the dynamic window approach to collision avoidance. In *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1280–1287, 1996.

[20] Masato Abe. *Motion and Control of a Vehicle (in Japanese)*. Tokyo Denki University Press, 2 edition, 2012.

[21] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. ROS : an open-source Robot Operating System. In *Open-Source Software workshop of the International Conference on Robotics and Automation*, 2009.

[22] Yohei Kakiuchi, Kunio Kojima, Eisoku Kuroiwa, Masaki Murooka, Shintaro Noda, Iori Kumagai, Ryohei Ueda, Fumihito Sugai, Shunichi Nozawa, Kei Okada, and Masayuki Inaba. Development of humanoid robot system for disaster response through team nedo-jsk’s approach to darpa robotics challenge finals. In *Proceedings of the 2015 IEEE-RAS International Conference on Humanoid Robots*.

[23] DARPA Robotics Challenge Finals in DARPAtv. https://www.youtube.com/user/DARPAtv/featured.