Journal of Engineering and Applied Sciences 12 (14): 3788-3792, 2017
ISSN: 1816-949X

Semi-Automatic Control of Telepresence Robots

Dmitry Suvorov, Roman Zhukov, Dzmitry Tsetserukou, Anton Evmenenko and Boris Goldstein
Department of Robotics, Labortary of Space Robotics,
Skolkovo Institute of Science and Technology, Bauman Moscow State Technical University,
Moscow, Russia

Abstract: Companies all over the world started to produce and sell telepresence robots in the last decade. Users of these robots have to plan their way to avoid collisions and control the speed of the robot to move it to destination point. This interface is familiar to many people who play computer games but it causes difficulties for the older generation and disabled people. We present here a novel interface to control telepresence robots where the operator only specifies the destination point on the video frame and the robot moves there automatically and avoids collisions. Users of the robots learn how to perform tasks 7-28% faster and make 17-49% fewer mistakes. The approach can be easily applied for any commercial telepresence robot and for development of telepresence robots for disabled people.

Key words: Mobile robotics, telepresence, control systems, navigation, user-friendly interface

INTRODUCTION

Telepresence robots were one of the first robots that entered the mass market. They became the next generation of teleconferencing systems. Over the past decade, the quest to develop a user-friendly interface for telepresence robots (Do *et al*., 2013; Tsetserukou *et al*., 2007) and special interfaces for disabled people (Leeb *et al*., 2015) has motivated significant research efforts.

There are three categories of previously developed approaches to telepresence robot control. People have to use keyboard or joystick to control translational and rotational speeds of the robot when the robot has the first category interface. The drawback of this approach is that users have to plan robot's trajectories manually and to avoid collisions with dynamic objects. Researchers solved it in the second approach where the operator specifies the destination point for the robot on the map of the environment using mouse or touch screen (Kwon *et al*., 2010). Now the robot cannot operate in an unfamiliar place because it needs the environment map. The robot can build a map using LIDAR, video cameras or their combination. The third approach combines different methods of robot control for disabled people. For example, researchers have tried to combine the first approach with brain computer interface.

In this study, we demonstrate a new approach to telepresence robot control. It is based on the kinematic and optical model of robot and on a purely reactive approach to the navigation problem (Blanco *et al*., 2008). Therefore, the robot can operate in an unfamiliar place. Operators only specify destination point on the video frame and the robot travels there automatically. The purely reactive approach to the navigation problem was originally designed for autonomous mobile robots with precise localization system, for example, based on LIDAR, odometry and particle filtering. We demonstrate here that it can extend the user interface of telepresence robots without precise localization system.

MATERIALS AND METHODS

Overview of the approach: All telepresence robots have at least one video camera. Some of them have LIDAR and odometry system installed onboard. We shall consider only such robots here. The operator has to specify destination point P^F_d on the video frame in the proposed approach. Next, the system needs to convert the image coordinates of the destination point into the relative coordinates of the robot's platform P^R_d. The last step is automatic travel of the robot to the destination point. The optical model of the robot's camera can be described as shown in Eq. 1:

$$\begin{pmatrix} x \\ y \\ \omega \end{pmatrix} = \begin{pmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{pmatrix} . \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{1}$$

Corresponding Author: Dmitry Suvorov, Department of Robotics, Labortary of Space Robotics,
Skolkovo Institute of Science and Technology, Bauman Moscow State Technical University,
Moscow, Russia

It is necessary to expand Eq. 1 to take into account the distortion of the lens. We get Eq. 2:

$$\begin{cases} x' = x\cdot\left(1+k_1\cdot r^2+k_2\cdot r^4+k_3\cdot r^6\right)+ \\ 2\cdot p_1\cdot x\cdot y+p_2\cdot\left(r^2+2\cdot x^2\right) \\ y' = y\cdot\left(1+k_1\cdot r^2+k_2\cdot r^4+k_3\cdot r^6\right)+ \\ p_1\cdot\left(r^2+2\cdot y^2\right)+2\cdot p_2\cdot x\cdot y \end{cases} \tag{2}$$

Where:

$(x, y)^T$ = The coordinates of the projection of the point on the camera's matrix with ideal lens

$(x', y')^T$ = The coordinates of the projection of the point on the camera's matrix with real lens

$(X, Y, Z)^T$ = The coordinates of the point in the camera's coordinate system

f_x, f_y = Focal lengths

c_x, c_y = The coordinates of the optical center of the cameras lens on the camera's matrix

$(k_1, k_2, k_3)^T$ = Radial distortion coefficients

$(p_1, p_2)^T$ = Tangential distortion coefficients

where, f_x, f_y, c_x, c_y, $(k_1, k_2, k_3)^T$, $(p_1, p_2)^T$ are determined using the procedure of camera calibration. When the user specifies $(X', y')^T$ using mouse or touch screen to move the robot there. there is an infinite number of $(X, Y, Z, 1)^T$ which can be the solution of Eq. 1. All these solutions lie on the same line L. This line L passes through the optical center of the camera that has the coordinates of $P_0 = (0, 0, 0, 1)^T$ in the camera's coordinate system. Another point of the line $P_1 = (X, Y, Z, 1)$ can be calculated from Eq. 1 and Eq. 2 assuming Z = 1.

Next, we need to transform these points from the coordinate system associated with the camera to the coordinate system associated with the base of the robot to use them for calculation of the real destination point. So, we need to describe the camera position using Denavit-Hartenberg parameters and calculate transition matrix T $(q_1 \ldots q_n)$ (Craig, 1955). The coordinates of P_0 and P_1 in the coordinate system associated with the robot's base are as in Eq. 3 and 4:

$$P_0^R\left(q_1 \ldots q_n\right) = T\left(q_1 \ldots q_n\right).\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{3}$$

$$P_1^R\left(x,\ y,\ q_1\ \ldots\ q_n\right) = T\left(q_1 \ldots q_n\right)\times P_1(x,y) \tag{4}$$

The parametric form of the equation of line L (Eq. 5):

$$P(t) = P_0^R + t\times\left(P_1^R - P_0^R\right) \tag{5}$$

We need to assume that Z = 0 in Eq. 5 to calculate destination point P^R_d in the coordinate system associated with base of the robot.

We use reactive navigation system described by Blanco *et al*. (2008) to move the robot to destination point P^r_d. We use odometry as a localization system.

Implementation: The approach was implemented for the telepresence Robot Webot (Fig. 1) equipped with the RPLidar laser scanner. Its kinematic chain configuration described using Denavit-Hartenberg parameters is shown in Fig. 2. Equation 6-9 demonstrate the calculation of transition matrix T:

$$A_1 = \begin{pmatrix} 1 & 0 & 0 & a_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{6}$$

$$A_2(q2) = \begin{pmatrix} \cos q_2 & 0 & \sin(q_2) & a_2\times\cos(q_2) \\ \sin(q_2) & 1 & -\cos(q_2) & a_2\times\sin(q_2) \\ 0 & 0 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{7}$$

$$A_3(q3) = \begin{pmatrix} \cos(q_3) & -\sin(q_3) & 0 & a_3\times\cos(q_3) \\ \sin(q_3) & \cos(q_3) & 0 & a_3\times\sin(q_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{8}$$

$$T(q_2, q_3) = A_1\times A_2(q_2)\times A_3(q_3) \tag{9}$$

Equation 1 was written for the case when axis Z and the optical axis of the camera match. Axis Y matches with the optical axis in our case. So, we updated equation 1 to arrive at Eq. 10:

$$\begin{pmatrix} x \\ y \\ \omega \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}.\begin{pmatrix} -Z \\ -X \\ Y \end{pmatrix} \tag{10}$$

We implemented an algorithm using C++ language and OpenCV library (Bradski and Kaehler, 2008). We used implementation of reactive navigation system from mrpt library. Our system was able to work in real time on arm7 processor. Also, we implemented user interface for web browser with the help of html5 (Fig. 3).

Fig. 1: Telepresence Robot Webot

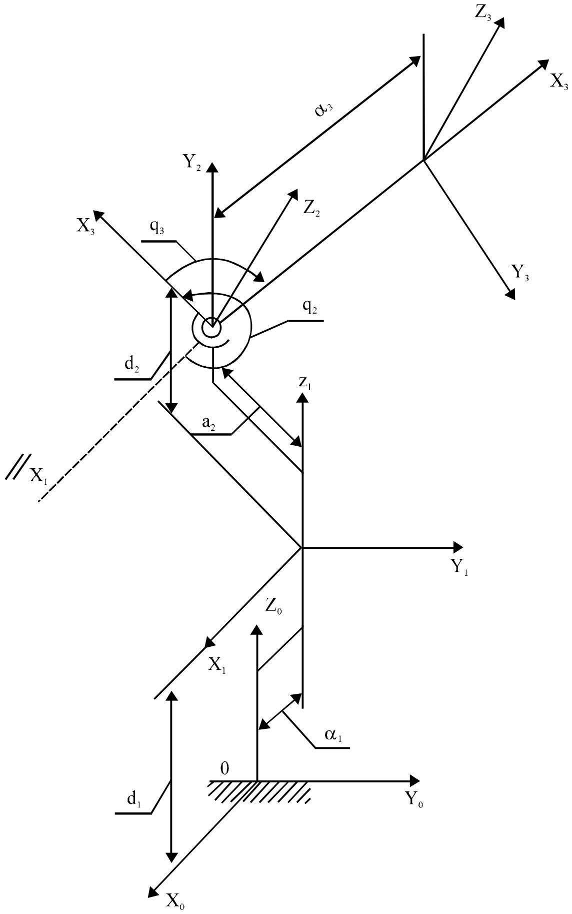

Fig. 2: Webot kinematic chain

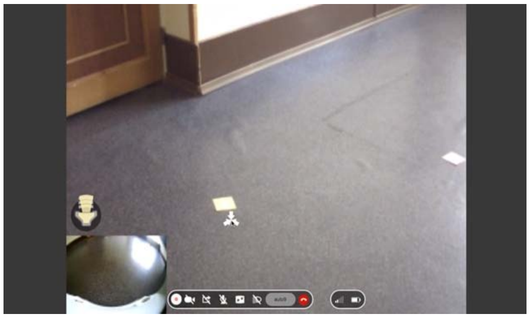

Fig. 3: Webot user interface

RESULTS AND DISCUSSION

Webot experiments: In the first experiment webot, controlled alternately by an operator and by the algorithm we designed was supposed to travel 2 m over an open space and stop in a specially marked area (1×0.6 m). Ideally, the robot had to stop in such a way that the center of its body matched the center of the area and the robot retained its spatial angle. During the experiment we measured deltas x, y, f (the distances from the center of the marked area along the corresponding axes and the angle in this coordinate system) and the travel time. These data are shown in Table 1 and 2.

In the second experiment webot, controlled alternately by an operator and by the designed algorithm was supposed to travel 2 m through a doorway and stop in a specially marked area (1×0.6 m). Ideally, the robot had to stop in such a way that the center of its body matched the center of the area. Similar measurements were taken but the angle value was omitted as irrelevant. The data are demonstrated in Table 3 and 4.

In the third experiment webot, controlled alternately by an operator and by the designed algorithm was supposed to travel 2 meters, bypassing a 1×1 m block and stop in a specially marked area (1×0.6 m). Ideally, the robot had to stop in such a way that the center of its body matched the center of the area. We took the same measurements as in experiment 2. The data are presented in Table 5 and 6.

As a result of the experiments we calculated the percentage that showed the advantage of the control method we designed over a person in solving such tasks. The criteria for the best results were delta values approaching zero (precision of axis and angle positioning) and shorter travel time. The intermediate and final (general) results are shown in Table 7 and 8 (the final ones are highlighted). The positive figures in the final results of the experiment demonstrate clear advantage of the designed algorithm in controlling the robot.

Table 1: Open space, operator control

x (cm)	y (cm)	f (rad)	t (sec)
-6.8	-9.2	-0.44	10.1
5.2	12.3	-0.73	9.2
15.6	13.9	-0.44	9.5
3.8	11.3	-0.16	9.2
-7.9	-7.5	0.27	7.5
-6.2	-6.1	-0.08	9.4
-1.8	3.8	0.16	9.1
-3.1	-11.0	0.11	8.9
6.7	2.1	-0.59	8.4
-12	3.8	-0.17	8.2

Table 2: Open space, designed algorithm control

x (cm)	y (cm)	f (rad)	t (sec)
2.8	11.8	-0.38	8.7
1.5	5.4	-0.06	8.9
-2.4	-12	-0.35	7.7
-3.8	-2.7	-0.27	8.1
-3.5	4.6	-0.38	6.8
0.1	3.7	-0.22	8.9
9.5	1.9	-0.18	7.9
-1.8	-13.3	0.11	9.3
-10	-9.5	0.05	8.1
0.3	2.7	0.23	8.6

Table 3: Doorway, operator

x (cm)	y (cm)	t (sec)
-8.3	-18.0	10.9
8.9	-9.2	12.3
10.1	14.4	10.4
11.4	4.6	11.3
0.6	5.2	10.3
2.2	5.5	10.7
-5.6	3.9	9.4
-8.1	1.5	12.2
5.0	-9.3	10.9
5.2	-2.7	13.3

Table 4: Doorway, designed algorithm control

x (cm)	y (cm)	t (sec)
-0.41	-0.7	10.8
8.9	-7.1	8.8
-2.2	-1.1	10.4
3.1	-6.4	9.0
-14.3	7.8	8.9
-2.2	1.1	9.5
-8.2	-9.8	9.3
-9.6	-9.6	9.5
-0.3	6.9	7.0
-2.9	-3.2	8.2

Table 5: Block, operator

x (cm)	y (cm)	t (sec)
-1.5	-7.3	13.8
1.4	-11.2	16.9
-9.3	-3.4	19.1
-6.9	7.4	16.7
-7.2	30.9	17.6
-2.5	13.1	17.8
-7.9	-5.1	20.3
5.1	-6.6	19.7
6.0	-4.2	15.5
-6.7	-5.9	14.5

Table 6: Block, designed algorithm

x (cm)	y (cm)	t (sec)
-4.0	5.4	17.7
-2.7	-5.7	18.9
-3.6	-1.7	10.9
-6.4	-1.2	11.5
3.0	3.7	8.9
5.3	-10.1	9.2
-3.7	0.0	10.5
-3.2	3.6	10.1
-0.2	9.8	13.3
0.9	8.1	13.0

Table 7: Open space

Variables	x (%)	y (%)	f (%)	t (%)
1	57	-32	18	16
2	53	85	206	3
3	192	23	28	20
4	6	106	-34	12
5	63	36	-34	8
6	88	30	-43	6
7	-110	23	-6	13
8	19	-28	31	-4
9	-47	-91	166	3
10	168	14	-18	-4
Results	49	17	31	7

Table 8: Doorway

Variables	x (%)	y (%)	t (%)
1	121	233	1
2	0	28	32
3	121	179	0
4	127	-24	21
5	-209	-35	13
6	0	59	11
7	-40	-79	1
8	-23	-109	25
9	72	32	35
10	35	-7	46
Results	20	28	18

CONCLUSION

We demonstrate here that it can extend the user interface of telepresence robots without precise localization system.

ACKNOWLEDGEMENTS

This research was assisted by the Wicron company. It received grant support from the Skolkovo Foundation. The researcher applied for the Russian patent ‘Hybrid method of telepresence’ (application 08071. date 08.12.2015).

REFERENCES

Blanco, J.L., J. Gonzalez and J.A.F. Madrigal, 2008. Extending obstacle avoidance methods through multiple parameter-space transformations. Auton. Robots, 24: 29-48.

Bradski, G. and A. Kaehler, 2008. Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media Inc., Sebastopol, CA., USA., ISBN-13: 9780596516130, Pages: 575.

Craig J.J., 1955. Introduction to Robotics: Mechanics and Control. Addison-Wesley, Boston, Massachusetts,.

Do, H.M., C.J. Mouser, Y. Gu, W. Sheng and S. Honarvar *et al*., 2013. An open platform telepresence robot with natural human interface. Proceedings of the 2013 IEEE 3rd Annual International Conference on Cyber Technology in Automation Control and Intelligent Systems (CYBER'13), May 26-29, 2013, IEEE, Nanjing, China, ISBN:978-1-4799-0611-6, pp: 81-86.

Kwon, O.H., S.Y. Koo, Y.G. Kim and D.S. Kwon, 2010. Telepresence robot system for English tutoring. Proceedings of the 2010 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO'10), October 26-28, 2010, IEEE, Seoul, South Korea, ISBN:978-1-4244-9122-3, pp: 152-155.

Leeb, R., L. Tonin, M. Rohm, L. Desideri and T. Carlson *et al*., 2015. Towards independence: A BCI telepresence robot for people with severe motor disabilities. Proc. IEEE., 103: 969-982.

Tsetserukou, D., R. Tadakuma, H. Kajimoto, N. Kawakami and S. Tachi, 2007. Development of a whole-sensitive teleoperated robot arm using torque sensing technique. Proceedings of the 2nd Joint EuroHaptics Conference and World Haptics Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, March 22-24, 2007, IEEE, Tsukaba, Japan, ISBN:0-7695-2738-8, pp: 476-481.