23rd ABCM International Congress of Mechanical Engineering
December 6-11, 2015, Rio de Janeiro, RJ, Brazil

A SERVOCONTROLLED 3-RRR PARALLEL ROBOR OPERATION MODES AND WORKSPACE SAFETY

Guilherme de Faveri
Clodoaldo Schutel Furtado Neto
Alexandre Campos
Universidade do Estado de Santa Catarina (UDESC), Centro de Ciências Tecnológicas (CCT), Joinville, Brazil
guilherme.faveri@gmail.com
clo_neto@msn.com
aac.campos@gmail.com

Abstract. *Robotic manipulators have been used in industry in different application such pick and place, assembly and welding. Based in closed kinematic chains, parallel robots presents high relation between stiffness and weight, otherwise this architecture presents limited workspace and different kinds of singularity. This paper presents some algorithms to control a servomotor based 3-RRR parallel robot. These algorithms control on-line a master-slave replication using computer mouse as master and 3-RRR parallel robot as slave. The position system control uses the internal servomotor control which presents satisfactory results in an Arduino based application. Developed algorithms controls position through solution of Inverse Kinematic Problem employing a workspace parametrization to skip from inverse kinematic singularity.*

Keywords: *Inverse Kinematic Problem, Parallel Robot, Singularity Deviation, On-line control*

1. INTRODUCTION

Since Industrial Revolution it have been an increasing demand to increase productivity rate and reduce costs (Tsai, 1999). In manufacturing terms, a second industrial revolution have been occurred since the automation was introduced.

Due to the high cost and inflexibility of heavy automation, when every machine is designed to a specific function, a different approach was developed. Robot manipulator was introduced aiming at performing tasks like point-to-point welding, painting and assembly. This robotic manipulators are controlled by computers or microcontroller which can make this machine reprogrammable (Tsai, 1999).

Parallel robot, or parallel manipulator, typically consist of a moving platform connected to a fixed base through several links which some joints are usually passive. Formally, a parallel robots are closed kinematic chain mechanisms which the end effector is linked to the base by independent kinematic chains (Merlet, 2002).

In general, mobility in parallel mechanisms is determined by the number of kinematic chains. In other terms, the degrees of freedom number is the same as the number of kinematic chains (Merlet, 2002). The robot used in this study is a 3-RRR planar parallel robot. It is compound by three kinematic chains which there are two links in every chain with three rotational joints. The first rotational joint, attached to the base, is the actuated one in every kinematic chain. This parallel architecture is well known and it is easy to a prototype building.

A master-slave control system is based in a master side where a operator perform tasks in an input device and a slave side where a robot perform the correspondent task (Horise *et al.*, 2012). Master-slave systems are present in several application, specially in medical industry like minimal invasive surgery (Oh *et al.*, 2013) and single-incision laparoscopic surgery (Horise *et al.*, 2012). Master-slaves control systems are largely found in systems which is required teleoperation.

Teleoperated systems are useful when a general-purpose system capable of doing several different tasks are required (Sabater *et al.*, 2006). Several teleoperated robots have been designed for specific tasks that needs teleoperation, such exploration of dangerous environments (Welch and Edmonds, 1993) and waste storage tanks repair (Schempf, 1995).

2. INVERSE KINEMATIC PROBLEM

The inverse kinematic problem consists in determine the joints variable value to a known position and orientation of end effector. The solution to this problem is quite important in order to transform the end effector movement to the movement of active joints that perform the imposed movement (Siciliano *et al.*, 2009).

Due to non-linearity of solution equation and possibility of multiply solutions the inverse kinematic problem may be very complex. Also, the solution existence is only guaranteed in robot workspace (Siciliano *et al.*, 2009).

To a 3-RRR planar parallel robot the inverse kinematic problem consist in determine the actuator angles θ_1, θ_2, θ_3

between the crank and coordinated axis x for a given position P and orientation φ. A schematic designed is presented in Fig. 1. The vectorial equations describing the fixed base are presented in Eq. 2, Eq. 3 and Eq. 3.

$$
\begin{aligned}
{}^f\vec{OA_1} &= 0\,\hat{i} + 0\,\hat{j} &(1)\\
{}^f\vec{OA_2} &= L\,\hat{i} + 0\,\hat{j} &(2)\\
{}^f\vec{OA_3} &= \frac{L}{2}\,\hat{i} + \frac{L\sqrt{3}}{2}\,\hat{j} &(3)
\end{aligned}
$$

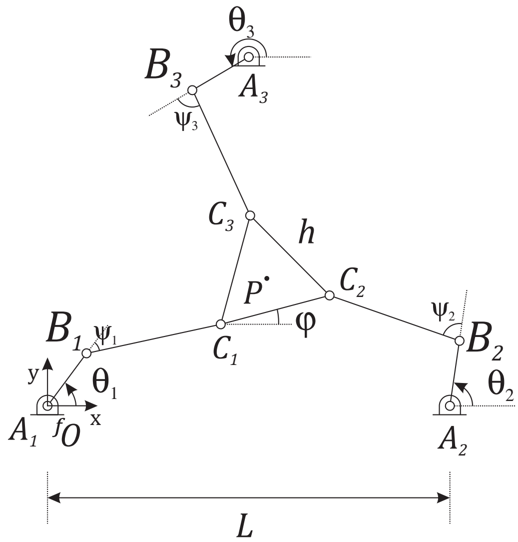


Figure 1. 3-RRR planar parallel robot schematic design.

For a given end effector configuration $T\vec{C}P = [\ P_x\ P_y\ \varphi\ ]$, the C_i point position is given by Eq. 4. In this way, kinematic chain equation to the joint $i = 1$ is presented in Eq. 5.

$$ {}^f\vec{OC_i} = {}^f\vec{OP} - \vec{PC} \tag{4} $$

$$ {}^f\vec{OC_1} = \left(P_x - \frac{h\sqrt{3}}{3}\cos\left(\varphi + \frac{\pi}{6}\right)\right)\hat{i} + \left(P_y - \frac{h\sqrt{3}}{3}\sin\left(\varphi + \frac{\pi}{6}\right)\right)\hat{j} \tag{5} $$

In order to shorten other kinematic chain equation, two constants C_{1_x} and C_{1_y} are determined in Eq. 6 and Eq. 7 to a given end effector configuration. This way, next kinematic chains can be written as presented in Eq. 9 and Eq. 9.

$$ C_{1_x} = \left(P_x - \frac{h\sqrt{3}}{3}\cos\left(\varphi + \frac{\pi}{6}\right)\right) \tag{6} $$

$$ C_{1_y} = \left(P_y - \frac{h\sqrt{3}}{3}\sin\left(\varphi + \frac{\pi}{6}\right)\right) \tag{7} $$

$$
\begin{aligned}
{}^f\vec{OC_2} &= \left(C_{1_x} + h\cos\varphi\right)\hat{i} + \left(C_{1_y} + hsin\varphi\right)\hat{j} &(8)\\
{}^f\vec{OC_3} &= \left(C_{1_x} + h\cos\left(\frac{\pi}{3} + \varphi\right)\right)\hat{i} + \left(C_{1_y} + h\sin\left(\frac{\pi}{3} + \varphi\right)\right)\hat{j} &(9)
\end{aligned}
$$

In order to find the angles θ_1, $\theta 2$ and θ_3 the closed loop equation presented in Eq.10 must be solved. To this, some algebraic manipulation are taken to eliminate the presence of passive joint angle ψ_i in vectorial equations that describe $\vec{BC_i}$. These steps are presented in Eq. 11 to Eq.15

$$ {}^f\vec{OA_i} + \vec{A_iB_i} + \vec{B_iC_i} + \vec{C_i\,{}^fO} = \vec{0} \tag{10} $$

$$ \vec{A_iB_i} = (a_i\cos\theta_i)\,\hat{i} + (a_i\sin\theta_i)\,\hat{j} \tag{11} $$

$$\vec{B_I C_i} = (b_i \cos(\theta_i + \psi_i))\,\hat{i} + (b_i \sin(\theta_i + \psi_i))\,\hat{j} \tag{12}$$

$$x_C - a_i \cos\theta_i = b_i \cos(\theta_i + \psi_i) \tag{13}$$

$$y_C - a_i \sin\theta_i = b_i \sin(\theta_i + \psi_i) \tag{14}$$

$$x_{C_i}^2 + y_{C_i}^2 + a_i^2 - b_i^2 - 2x_{C_i} a_i \cos\theta_i - 2y_{C_i} a_i \sin\theta_i = 0 \tag{15}$$

Aiming at solving Eq. 15, the parameters e_{1_i}, e_{2_i} and e_{3_i} are introduced and the classical Weierstrass substitution is used to reduce this equation to form presented in Eq.20 (Tsai, 1999).

$$e_{1_i} = x_{C_i}^2 + y_{C_i}^2 + a_i^2 - b_i^2 \tag{16}$$

$$e_{2_i} = -2x_{C_i} a_i \cos\theta_i \tag{17}$$

$$e_{3_i} = -2y_{C_i} a_i \sin\theta_i \tag{18}$$

$$e_{1_i} + e_{2_i} \cos\theta_i + e_{3_i} \sin\theta_i = 0 \tag{19}$$

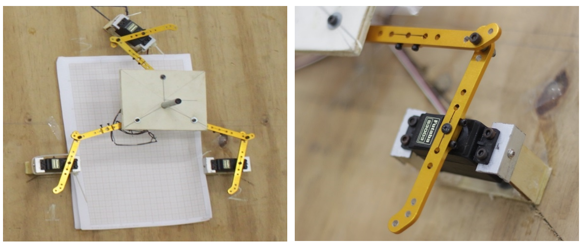
$$(e_{1_i} - e_{3_i})\, t_i^2 + 2e_{3_i} t_i^2 + (e_{1_i} + e_{2_i}) = 0 \tag{20}$$

Thereby, is it possible to solve the quadratic equation to the variable t_i which is defined by the Weierstrass tangent half-angle substitution as $t_i = \frac{\theta_i}{2}$. The closed form solution to the inverse kinematic problem is given by Eq. 21 to the kinematic chain $i = 1, 2, 3$.

$$\theta_i = 2 \arctan\left(\frac{-e_{3_i} \pm \sqrt{e_{3_i}^2 + e_{2_i}^2 - e_{1_i}}}{e_{1_i} - e_{2_i}}\right) \tag{21}$$

3. ROBOT DESIGN

A 3-RRR planar parallel robot was built to test some position control algorithm. The fixed base consists to a wood plate which is attached three R/C servomotors ordinarily used in model airplane. Three Futaba S3003 servomotors were chosen due to easily control, interfacing with Arduino and high rate between torque and mass. To compose the kinematic chain, bolts and nuts were used as pin in rotative joints. The built robot is presented in Fig. 2 and design parameters are shown in Tab. 1



(a) (b)

Figure 2. 3-RRR built in (a) and detail in kinematic chain and active joint in (b)

Table 1. Robot design variables

Part	Dimension	Length (mm)
Rod	b	88.90
Crank	a	44.45
Base	L	0.26
end effector	l	85.73

4. ROBOT CONTROL AND OPERATION MODES

To properly control the parallel robot, it was necessary to verify the robot workspace. It is known that the 3-RRR parallel do not present singularity into it's workspace (Tsai, 1999) as presented in Fig. 4. The workspace is obtained by successively solving the Eq. 20 to an orientation $\varphi = 0$ in a simulation developed in Scilab.

Aiming at performing communication between entry data system and the robot, serial port and Arduino servo library was used. A software to perform integration between Arduino and user was developed in order to create a master/slave on-line control system using mouse cursor as master data entry in Processing IDE. This software perform transformation from graphic user's interface to the real robot workspace and write in serial port the angle which must be attached to active joints using Eq. 22, which v is the value to be attached to servomotor and ch is the character read by Arduino software in serial port. If ch is not a number but a letter, v is attached to a servomotor identified by the letter.

$$v = 10v + ch \tag{22}$$

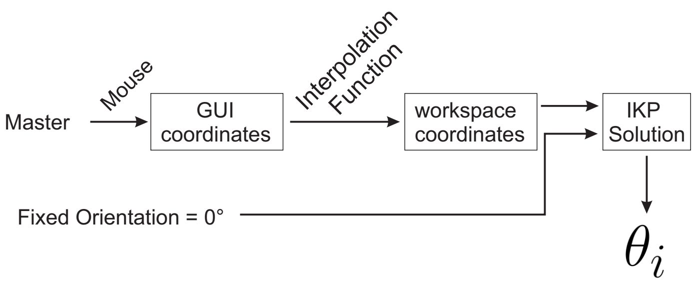


Figure 3. Programming schematics to robot control.

In order to convert virtual workspace to the robot workspace, nine points, p_i (m), are taken inside the robot workspace and coordinates are associated to nine points in virtual workspace, m_i (pixels). Taken points are shown in Fig. 4 and Fig. 5 and coordinates are shown in Tab. 2.

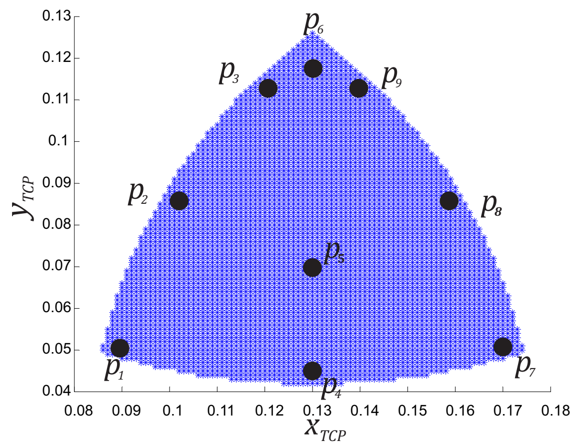


Figure 4. 3-RRR workspace to an orientation $\varphi = 0$.

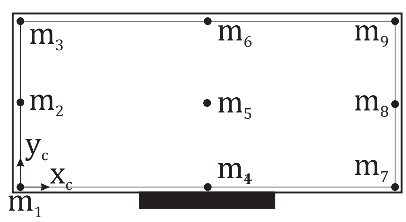


Figure 5. Points taken in GUI.

Table 2. Taken points in GUI and correspondents in robot workspace

GUI	(x_c, y_c) (pixels)	Workspace point	(x_{TCP}, y_{TCP}) (m)
m_1	(0,0)	p_1	(0.090,0.050)
m_2	(0,357)	p_2	(0.098,0.100)
m_3	(0,715)	p_3	(0.122,0.115)
m_4	(655,0)	p_4	(0.130,0.045)
m_5	(655,357)	p_5	(0.130,0.070)
m_6	(655,715)	p_6	(0.130,0.120)
m_7	(1310,0)	p_7	(0.170,0.050)
m_8	(1310,357)	p_8	(0.150,0.100)
m_9	(1310,715)	p_9	(0.138,0.115)

4.1 Linear Interpolation

The first operation mode consists in use a linear function to interpolate between virtual workspace coordinates and the real workspace coordinates. Proposed equations are independent and presented in Eq. 23 and Eq. 24.

$$x_{TCP} = f\left(x_c\right) = C_{Lin_1} + C_{Lin_2} x_c \tag{23}$$

$$y_{TCP} = f\left(y_c\right) = C_{Lin_4} + C_{Lin_4} y_c \tag{24}$$

Two points are taken in each direction to solve the equations systems and determine the coefficients. For coordinates in x direction, points are m_1 and m_7 that correspond to p_1 and p_7 respectively. To y direction, taken points are m_4 and m_6 corresponding to p_4 and p_6 respectively. Resulting coefficients are shown in Tab. 3

Table 3. Constants to linear interpolation functions

Constant	Value
C_{Lin_1}	0.090
C_{Lin_2}	$6.109E-5$
C_{Lin_3}	0.045
C_{Lin_4}	$1.049E-4$

4.2 Quadratic Interpolation

The second approach uses separation of x and y coordinates and the transfer functions are quadratic to both variable. Quadratic transfer functions are proposed to interpolate from cursor position x_c and y_c in graphic user's interface to real workspace describing the position of tool center point x_{TCP} and y_{TCP}. Equations 25 and 26 are the proposed and the constants C_1, C_2, ..., C_6 must be found.

$$x_{TCP} = f\left(x_c\right) = C_1 + C_2 x_c + C_3 x_c^2 \tag{25}$$

$$y_{TCP} = f\left(y_c\right) = C_4 + C_5 y_c + C_6 y_c^2 \tag{26}$$

Defining six points in user interface and six correspondent points in robot workspace, constants are found by solving the equation system Eq. 28 to Eq. 29 and Eq. 31 to Eq. 32. The points taken in GUI are m_1, m_4 and m_7 for x coordinate and m_4, m_5 and m_6 for y in Fig. 5. The correspondent points taken in robot workspace are respectively p_1, p_4 and p_7 for x coordinate and p_4, p_5 and p_6 for y coordinate from Fig. 4. This way, to any point in GUI a correspondent point is obtained to the robot. The resulting coefficients are shown in Tab. 4.

$$x_{TCP_1} = C_1 + C_2 x_{c1} + C_3 x_{c_1}^2 \tag{27}$$

$$x_{TCP_2} = C_1 + C_2 x_{c_2} + C_3 x_{c_2}^2 \tag{28}$$

$$x_{TCP_3} = C_1 + C_2 x_{c_3} + C_3 x_{c_3}^2 \tag{29}$$

$$y_{TCP_1} = C_1 + C_2 y_{c_1} + C_3 y_{c_1}^2 \tag{30}$$

$$y_{TCP_2} = C_1 + C_2 y_{c_2} + C_3 y_{c_2}^2 \tag{31}$$

$$y_{TCP_3} = C_1 + C_2 y_{c_3} + C_3 y_{c_3}^2 \tag{32}$$

Table 4. Constants to quadratic interpolation functions

Constant	Value
C_1	0.090
C_2	$6.109E-5$
C_3	0
C_4	0.045
C_5	$3.7762E-5$
C_6	$9.389E-8$

4.3 Biquadratic Interpolation

A third approach use biquadratic functions presented in Eq. 33 and Eq. 34 to transfer GUI to workspace. This approach consists in determine two biquadratic functions to both coordinates, in such way that robot x_{TCP} and y_{TCP} depend simultaneously of both GUI coordinates x_c and y_c. In this approach, two equation system of nine equations are solved and found the 18 constants presented in Tab. 5.

$$f(x_c, y_c) = C_1 + C_2x_c + C_3y_c + C_4x_cy_c + C_5x_c^2 + C_6y_c^2 + C_7x_c^2y_c + C_8x_cy_c^2 + C_9x_c^2y_c^2 \quad (33)$$

$$g(x_c, y_c) = C_{10} + C_{11}x_c + C_{12}y_c + C_{13}x_cy_c + C_{14}x_c^2 + C_{15}y_c^2 + C_{16}x_c^2y_c + C_{17}x_cy_c^2 + C_{18}x_c^2y_c^2 \quad (34)$$

Table 5. Constants to biquadratic interpolation

Constant	Value	Constant	Value
C_1	0.090	C_{10}	0.050
C_2	$61069E-5$	C_{11}	$-1.5267E-5$
C_3	0	C_{12}	$1.8881E-4$
C_4	$5.1246E-8$	C_{13}	$-4.6976E-7$
C_5	$1.4611E-23$	C_{14}	$1.1654E-8$
C_6	$6.2595E-8$	C_{15}	$-1.3693E-7$
C_7	$-7.8239E-11$	C_{16}	$3.5859E-10$
C_8	$-1,6274E-10$	C_{17}	$7.1673E-10$
C_9	$1.0943E-13$	C_{18}	$-5.4713E-13$

Using found coefficients to programming the robot, robot workspace was drawn as presented in Fig. 6. This workspace is similar to the robot workspace to constant orientation $\varphi = 0$. The distance between extreme points in x coordinate presented an error of 6.66% in comparison to simulated and the distance of extreme points in y coordinates presented an error of 6.35%.

Figure 6. 3-RRR planar parallel robot workspace drawn in second approach

5. INTERPOLATION COMPARISON

In order to compare the proposed interpolation, is used as parameter the rate between points in GUI that lead to a reachable position in robot workspace and full GUI. There are an area of 1310x715 pixel in virtual workspace, in this way the rate, r is defined in Eq.35 where N is the number of reachable points. In Tab.6 are shown r for each interpolation. In Fig. 7 and Fig. 8 are shown in black the areas in GUI workspace not reached by the parallel robot.

$$r = \frac{N}{1310 \times 715} \tag{35}$$

Table 6. Rate value to interpolations

Interpolation	r
Linear	0.7423
Quadratic	0.8109
Biquadratic	0.9536

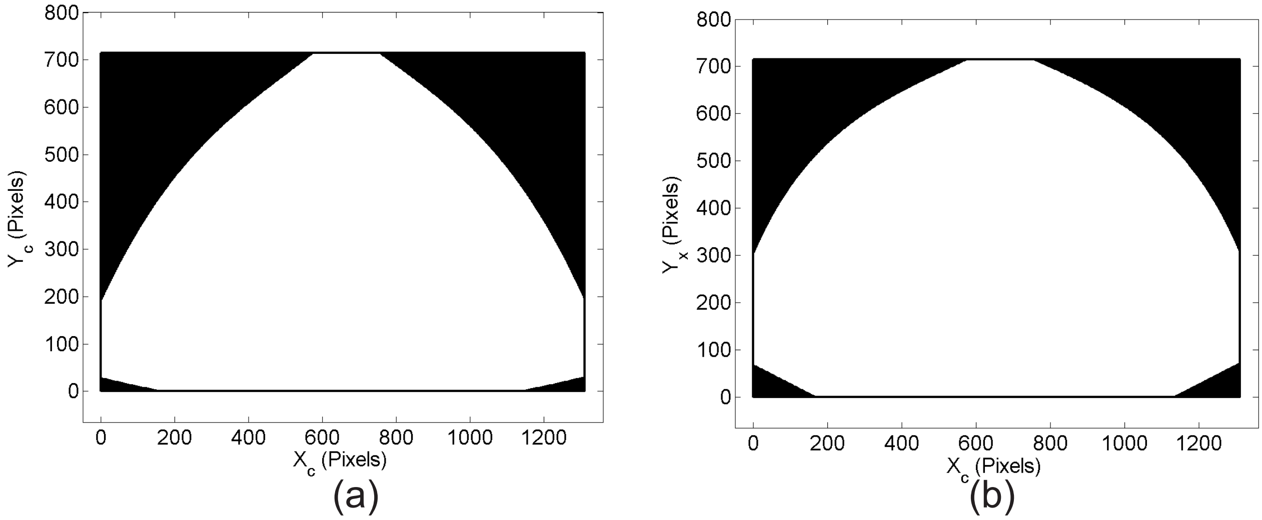


Figure 7. 3-RRR planar parallel robot workspace drawn in second approach

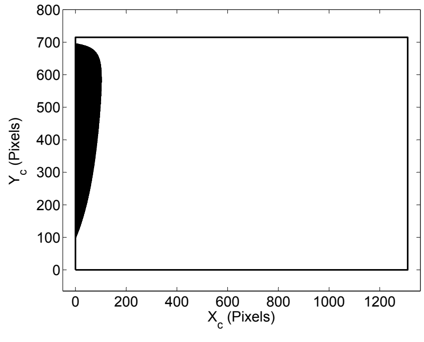


Figure 8. GUI not reachable by the parallel robot in biquadratic interpolation.

In Fig. 9 and Fig. 10 are shown two different master trajectory and slave trajectory, a linear and a circular track. It is possible to note that the biquadratic interpolation present a great distortion in track notwithstanding no singularity is reached.

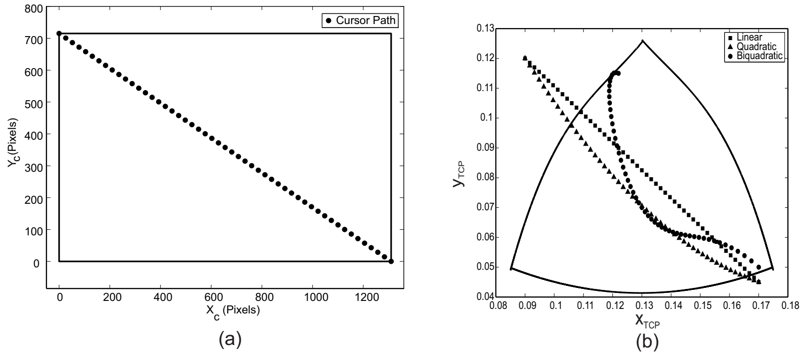


Figure 9. Linear track. (a) Master: Mouse movement (b) Slave: parallel robot movement

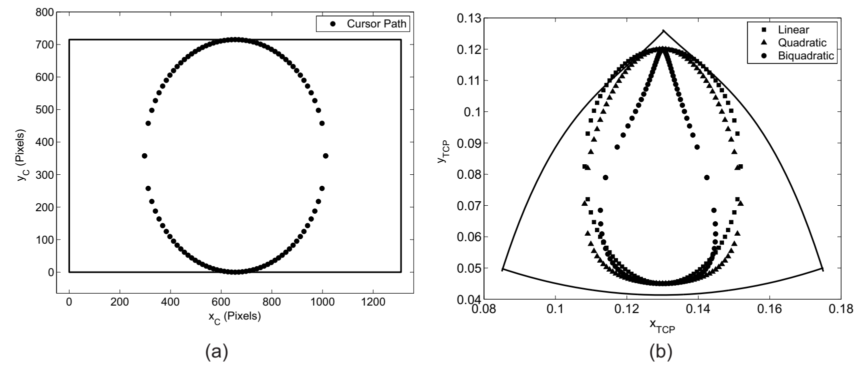


Figure 10. Circular track. (a) Master: Mouse movement (b) Slave: parallel robot movement

6. CONCLUSION

Using inverse kinematic problem solution to command the movement, a 3-RRR parallel robot is built. This robot is controlled by an Arduino reading the serial port using R/C servomotors as active joints. Due to internal servomotor control system, built robot presents a satisfactory control.

Three transfer functions to correlate a GUI in computer to real workspace was defined, one linear, one quadratic and another one biquadratic. The biquadratic approach presented a more safety algorithm since there are a greater number of reachable points, other wise it presented undesirable alterations in master/slave track control, generating curves in line trajectory and deforming curves.

7. REFERENCES

Horise, Y., Nishikawa, A., Sekimoto, M., Kitanaka, Y., Miyoshi, N., Takiguchi, S., Doki, Y., Mori, M. and Miyazaki, F., 2012. "Development and evaluation of a master-slave robot system for single-incision laparoscopic surgery". *International Journal of Computer Assisted Radiology and Surgery*, Vol. 7, No. 2, pp. 289–296.

Merlet, J.P., 2002. *Parallel Robots*. Springer.

Oh, J.S., Shin, W.K., Uhm, C.H., Lee, S.R., Han, Y.M. and Choi, S.B., 2013. "Control of haptic master Ű slave robot system for minimally invasive surgery (mis)". In *Journal of Physics: Conference Series*. Vol. 412.

Sabater, J.M., Saltarén, R.J., Aracil, R., Yime, E. and Azorín, J.M., 2006. "Teleoperated parallel climbing robots in nuclear installations". *Industrial Robot: An International Journal*, Vol. 33, No. 22, pp. 381–386.

Schempf, H., 1995. "Houdini: site and locomotion analysisdriven design of an in-tank mobile cleanup robot". In *American Nuclear Society Winter Meeting Transactions*.

Siciliano, B., Sciavicco, L. and Luigi Villani, G.O., 2009. *Robotics Modeling, Planning and Control*. Springer.

Tsai, L.W., 1999. *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. John WIley & Sons, Inc.

Welch, R. and Edmonds, G., 1993. "Applying robotics to hazmat". In *Proceedings of the Forth National Technology Transfer Conference Ű NASA Technology, Anaheim, CA, December*. Vol. 2, pp. 279–287.

8. RESPONSIBILITY NOTICE

The author(s) is (are) the only responsible for the printed material included in this paper.