EasyChair Preprint

№ 7801

Force Control Under Contact Motion of a Robotic Machining System

Usman Ahmad, Syed Irtiza Ali Shah, Malik Muhammad Awais and Muhammad Shaheer

April 18, 2022

FORCE CONTROL UNDER CONTACT MOTION OF A ROBOTIC MACHINING SYSTEM

Usman Ahmad
Department of Mechanical and Aerospace Engineering
Air University
Islamabad, Pakistan
190642@students.au.edu.pk

Syed Irtiza Ali Shah
Department of Mechanical and Aerospace Engineering
Air University
Islamabad, Pakistan
irtiza@students.au.edu.pk

Malik Muhammad Awais
Department of Mechanical and Aerospace Engineering
Air University
Islamabad, Pakistan
malik.awais19@gmail.com

Muhammad Shaheer
Department of Mechanical and Aerospace Engineering
Air University
Islamabad, Pakistan
190616@students.au.edu.pk

***Abstract*—The purpose of this project is to control the force under contact motion of a robotic machining system. Robot control is the system that controls the movement of different parts of robots and robotic machinery. it deals with the mechanical aspects and programming systems that make the control of robotic machinery possible. Force control enables robotic processes such as grinding, deburring, sanding, and machine polishing. when servicing the machine a torque sensor can help the robot to locate the stop of a fixture when it places the vice on a CNC machine. Force sensors also facilitate product testing packaging and robotic assembly applications. There are two methods to control force active and passive. the passive control system is an open-loop system and not using a way to adjust for force errors' on the other hand active force control system is a closed-loop system that can automatically adjust to force errors. The essential thought behind the force control is the output of the sensor is used to close the loop in the controller, altering each of the joint torque to coordinate the required output. Robots are a highly flexible tool for achieving quality machining. Despite their low stiffness as compared to CNC machines, robots are more capable of performing different types of machining tasks. but there is one issue hard materials are challenging for a piece of robotic machinery. When the tool contacts the surface of any hard material the robot can deflect and the drilled part becomes inaccurate but the force control is a good way to overcome this problem. By using force control robotic machining, machining operations become better because you can precisely control the force applied to the workpiece this improves the quality of drilling operation grinding operation. Force control under a contact motion in robotics machining is used in different applications different machining processes, automobile industries, aerospace industries.**

During last years, several control schemes have been proposed in the literature, such as pure force control, compliance control, impedance control, and hybrid control. Firstly, hybrid control is the only control strategy which allows simultaneous control of motion and force; the structure of this control system is modified by means of opportune selection matrices which are defined according to the task requirement. This paper presents s new force control method which may be thought as logically derived from compliance control and impedance control for which only a reference position is usually assigned. The attractive feature of the proposed control is that it also allows for a reference force to be assigned. In this way one may obtain analogous performance to hybrid control without adopting selection matrices. The advantage gamed with this choice lies in the achievement of a degree of robustness of the control scheme to inaccurate environment modelling. The parallel control approach has been developed starting from the analysis of the different components of the interaction. Each component has been characterized in terms of its function and relation to the others. This led to distinguish two different aspects in the overall control strategy.

Keywords— Robot, force, control, motion, freedom.

I. INTRODUCTION/ LITRATURE REVIEW

Using force control robotic machining, machining operations become better because you can precisely control the applied force on the workpiece.

Cleaning and pre-processing operations are important activities and represent a high-cost burden for cast manufacturers. Post-processing can cost an additional 40 percent of foundry production costs. Processing processes such as cleaning, milling, grinding, deburring, and sawing are promising applications for industrial robots powered by foundry automation. From a robot machining perspective, two types of machining processes could be distinguished. The first type, typically cleaning and deburring, typically has a very complex 3D curved cutting path, a critical cycle time requirement, and a relatively low surface accuracy requirement. Recently, most deburring operations are being performed manually in an extremely noisy, dusty, and unsanitary environment. Therefore, automation is highly desirable for these operations. The second type is the milling process, in which the robot travels a simpler path with a lower feed rate (20-30 mm / s), while the comes in close contact with the workpiece. The control must be precise enough to maintain the quality of the surface under large varying machining forces. This type of processing is currently performed by CNC machines, which is only economically justifiable for large batches. This study proposes a robotic machining strategy for the foundry industry with small to medium-size batches. The strategy is a complete solution that addresses the challenges for both types of machining applications, from programming to process control. This article addresses a series of important issues in the robotic machining process, from programming to process control. Three main contributions, namely easy robot programming,

inline deformation compensation, and CMRR, have been detailed. The complete solution is achieved with the force control strategy based on the ABB IRC5 robot controller

Industrial robots are increasingly used for machining jobs. The main reason is that the investment cost of buying these robots is much lower than the cost of buying a CNC machining center. Other advantages are the relatively small ground area required for construction, compared to the large workspace of the robot, and the fact that the robot can easily perform pre-and post-processing procedures. Therefore, it is not necessary to carry out additional equipment for loading and unloading machining centers. Of course, the comparison would not be complete without considering the limitations of the robot.

In a typical fixing application, the goal is to remove key burrs along part edges, e.g. from casting or machining, with a rotary cutting tool mounted on the main bearing. In addition, it is generally desirable to bend the edge at a 45˚ angle so that workers are not injured during handling and/or assembly. The application considered here is the repair task of water pump turbines of various shapes and sizes. The wheels are created from high-strength copper in a casting process with many further steps of traditional CNC machining. The burrs occur mainly along the edges of the impeller blades and, because of the measurable effect of the shape of the burrs on pump performance, should be carefully removed. This work has traditionally been performed by manual labor adapted to the high physical demands of workers in harsh environments. The experimental system is built around an industrial robot (ABB IRB 4400) with a maximum payload of 60 kg and equipped with a suite of force control applications. Attached to the robot flange, a torque sensor (ATI Omega 160) and a 3-finger gripper (SOMMER AUTOMATIC GD316) to unload the wheel directly from the CNC machine. The tabletop electric spindle actuator can provide speeds from 5,000 to 36,000 rpm with a maximum output power of 8 kW

The objective of subproject A6 at DFG's 708 Collaborative Research Center (German Research Association) at the Institute of Production Systems (IPS) of TU Dortmund is to develop a robot-based grinding process to Optimized geometry and surface properties of hard-coated surfaces. The objective of this research project is to model the grinding process with a robot, plan the robot trajectory based on process-adaptive simulation, and use force-controlled process execution. The objective of the project is to improve the robot process performance through the comprehensive and systematic optimization of the grinding process parameters and the robot control algorithms, especially those of the grinding process. robotic force control, to meet the requirements according to dimensional accuracy and surface quality. To verify the applicability of industrial robots in the grinding process with grinding heads, it is necessary, among other things, to examine the influence of the robotic system on the outcome of the machining process. One of the first steps is to form an industrial robot grinding model with grinding heads, describing the dependence of the removal rate on various process parameters (normal force, cutting speed, speed) feed, etc.). Such a model is intended to be used to predict the volume of material removed and the shape of the product after the process of machining.

In aircraft engines, gas turbines, steam turbines, turbine blades are the most important component. Their machining accuracy and surface quality determine the performance, operating efficiency, the service life of power equipment. These blades are difficult to cut because they are made up of heat-resisting steel titanium alloys and high-temperature alloys etc. these blades are designed for thin-walled twisted parts after different manufacturing and machining processes blades required superfinishing. For blade superfinishing, the current operation and research mainly focus on the two following means, manual finishing and abrasive belt grinding by multi-axis CNC machine tools. Compared with the blade body, the thin blade thickness and great curvature changes and machining path posture changes at the leading and trailing edges often results in difficulty in precision grinding of complex blades. Due to the randomness of positioning in the process of manual grinding, the removal allowance at the blade edge and the contact force between the grinder and the blade are difficult to be controlled, resulting in poor accuracy of waviness and cross-sectional shape, as well as a poor machining consistency between blades. While for multi-axis NC belt grinding, since the fixture positioning error of a blade in the magnitude is equivalent to the deformation error at the leading and trailing edges, thus CNC grinding must be based on an accurate measurement of the blade clamping. It is believed that the contact force between the robot end-effector and the external environment should be observed and controlled to better adapt to blade grinding. Therefore, many researchers have attempted meth-ods to estimate an unknown environment by F/T transducers, vision sensing, etc., and then a large number of works have been conducted from the angles of impedance control, hybrid force/position control.

In most industrial sectors, it is predominantly a manual operation where the human operators use hand tools to chamfer part edges, such as gears, cast parts, and surfaces. There is a significant need to automate the chamfering process not only to improve quality and performance but also to remove health hazards to human operators due to particulate/debris and ergonomic conditions. (2006); Pagilla and Yu (2001a) where the contact between robot and workpiece is usually surface contact, the contact between end-effector and workpiece is a line or even point contact in robotic chamfering. Additionally, most modern industrial robotic manipulators only give velocity or position control interfaces to the users and do not provide the ability to control motor torques for executing a trajectory.

When employing robot manipulators for chamfering, one of the commonly adopted methods is to generate a nominal trajectory from the CAD model of the workpiece and apply the position/orientation of the workpiece in the robot workspace. (2000) developed an automatic chamfering system using an industrial robot and applied it to chamfering of a hole by using the chamfering path generated from a CAD system. Registration is the process of obtaining the position of the workpiece in the specified coordinate frame by using tools from metrology that provide the accurate location/orientation of the workpiece. Indirect methods are used to modify the trajectories generated based on a CAD model instead of relying on metrology tools.

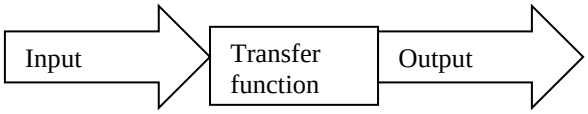
(2016) used point clouds registration to align the nominal CAD model with the actual workpiece, but the performance of this method largely depends on the accuracy of the vision system and the robustness of the registration algorithms. Song and Song (2013) tried to improve the CAD model-based trajectory generation in robotic deburring by registering the nominal trajectory with chosen points on the workpiece that are obtained from "teaching. " An alternative way to generate the trajectory was proposed by Zhang et al. Another major source of errors in the positioning and geometry of the work cell or fixture on which the workpiece is mounted in the robot workspace.

So, a flexible trajectory generation process that can adapt to the changes in the work cell structure is needed for accommodating workpieces with multiple geometries. The method proposed in this paper takes advantage of the high repeatability of industrial robot manipulators and generates a trajectory in line to avoid costly and time-consuming registration of the workpiece beforehand. The trajectory generated this way can also take into account the position change during gear mounting and avoid registration for each new gear mounted in the work cell.

As for machining, many studies have been reported and the results indicated that some critical issues, including trajectory error, material removal rate, and contacting force, are needed to be addressed. This statement is also agreed upon by Karayiannidis and Doulgeri , who proposed an adaptive leaning controller to identify the surface condition with the use of force and joint position/velocity measurements. However, the negative effects on machining performance, caused by the low stiffness of the robot, have not been discussed in these studies. since 1996, who studied the characteristics of robotic milling operation and proposed a method to reduce the effect caused by low stiffness.
identified the robot stiffness by two methods and conducted machining experiments whose results indicated that the path displacement, caused by robot low stiffness, plays a significant role in robotic machining error. present a compensation scheme based on the joint stiffness model to reduce the machining deformation caused by machining force. This paper presents a fuzzy-sliding mode control scheme to compensate for the deformation and oscillation caused by machining force. With the knowledge of machining behavior, a sliding mode controller is then constructed based on fuzzy rules to manage the complexity of robotic.

1. METHODOLGY

The external force is subjected when it is in contact with environment which experienced internal forces. In single vector loop variable only one position control or force control is presented. To overcome this the dynamic relationship between position and force variable is presented. To control force and position of robotic system we have used parallel approach. By using this approach force can be controlled and position by offering robustness to inaccurate environment. In the control scheme selection matrix was not used to obtain these feature but the force controller and position controller are combined by leaving the sole trajectory planner. The output force and position requirement are concern with desired input while the force controller overcome over the position controller. The subject which is already constrained to environment there is no need to control variable. The structure of control scheme is changed in a way to match the task structure by means of selection matrices. To avoid unwanted interface between two controllers the proper control action is active. For a clearly model based decision logic the force and position are evaluated. In parallel controller no information about location is required. Adoption of control strategies is demanded by constrained motion execution which can handle the interaction of manipulator with the environment and force and position is control properly. Rule-based controller has many advantages over model based controller with respect to both position and force controls.



Then from this diagram, equations from newton second law. Then taking Laplace and finding transfer function.

F - kx1 – dv1-k (x1-x2) – d (v1-v2) =m_1 a1

Putting the value of m_1=1,d=1and k=1

F(s) – x1 (s) – s x1 (s) – x1 (s) + x2 (s) – s x1 (s) + s x2 (s) = s^2 *x1 (s)

F(s) = x1 (s) [s^2 + s + s + 1 + 1] – x2 (s) – s x2 (s)

-k (x2 – x1) – d (v2 – v1) = m_2 a2

- x2 (s) + x1 (s) – s (x2 (s) – x1 (s)) = s^2 x2 (s)

-x2 (s) + x1 (s) – sx2(s) + sx1 (s) = s^2 x2 (s)

S^2 x2 (s) + s x2 (s) + x2 (s) = s x1 (s) + x1 (s)

X2 (s) [s^2 + s + 1] = x1 (s) [s + 1]

X2 (s) = [(s+1) / (s^2 + s + 1)] x1 (s)

F(s) = x1 (s) [s^2 + 2s + 2] – [s+1] [(s+1) / (s^2 + s + 1)] x2(s)

So transfer function is

G(s)=

$$\frac{X1\,(s)}{f(s)} = \frac{(s^2 + s + 1)}{[(s^2+2s+2)(s^2+s+1)-(s+1)^2]}$$

In the system design the main problem in parallel approach is that it allows to account dynamics and geometric

characteristics of the environment. To each component of control system this features is interesting. It has been proven that even the environmental stiffness is underestimated by an order of magnitude, the plan is satisfactory. Recover from an (unpredictable) abnormal state Mistakes in the plan lead to manipulator operation In spite of problems, a highly rigid environment can be achieved the saturation of the joint actuator torque.

2. IMPLEMENTATION

MATLAB and Simulink for analysis can be used for project. In MATLAB, calculations can be performed which cannot be done by hand. Sensor can be used to sense environment and for other various purposes such as PID.

A proportional integral derivative control is a control loop mechanism, and it is mostly used in industrial control system. To obtain a high accuracy of movement and to control end of robotics accurately PID controller is always used. it consists of derivative component, integral component, and proportional component. P controller, PI controller and PID controller can also be applied if needed.

Matlab is developed by math work and is a multi-programming language and numerical computing environment. Many engineering disciplines rely on various mathematics to ensure that the results of any design process or new theories about how the universe works are actually meaningful. If the new building cannot withstand the pressure applied to it, then it will be of little use. Numerical analysis depends on approximations, not precision as seen in symbolic mathematics. Without the application of numerical analysis, it is impossible to perform certain construction tasks, and astronomy seems to need to use it extensively because it seems to require heavy use of it as well.

Simulink is a graphical programming environment based on MATLAB for modeling, simulating and analyzing multi-domain dynamic systems. Its main interface is a graphical block diagram drawing tool and a set of customizable block library. It provides tight integration with the rest of the MATLAB environment, and can drive MATLAB or write scripts from it. Simulink is widely used in automatic control and digital signal processing for multi-domain simulation and model-based design.

A. Actuators

The main feature of actuator is the ratio of power to size. Recently, many actuator includes built-in position and speed sensors for measuring variables inside the robot and in the environment. The sensor is usually mounted on the motor shaft, not Joint axis. This type of smart actuator is based on sensors materials with sensors and actors have characteristics at the same time electromechanical subsystem. With such an actuator can accurately determine the output stroke and/or Output force without using other sensors stroke and strength. In this sense, the piezoelectric actuator is Very suitable for use as a smart actuator.

II. RESULTS AND ANALYSIS

A. For Non-contact

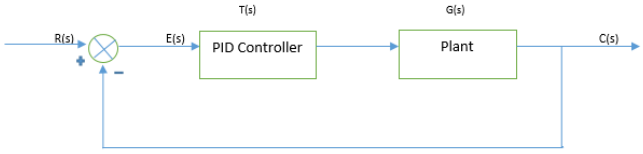


Fig. 1. Block diagram

Equations in time domain are as follows

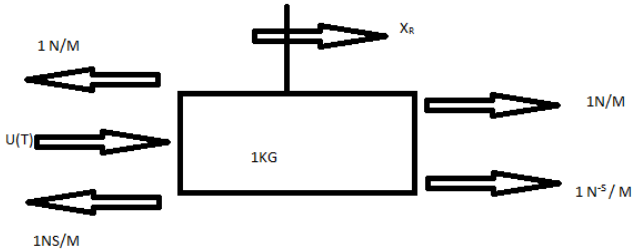


Fig. 2. FBD of mass 1

$u(t) - X_r - (X_r - X_s) - \dot{X}_r - (\dot{X}_r - \dot{X}_s) = \ddot{X}_r$

$u(t) - X_r - X_r + X_s - \dot{X}_r - \dot{X}_r + \dot{X}_s = \ddot{X}_r$

These are two equations for non-contact for 1st mass. In the first equation only one force is acting, and its displacement is denoted by X_r. two mass and two damper is connected to this system. And the values of all these coefficients is 1.

$\ddot{X}_r + 2X_r + X_s + 2\dot{X}_r + \dot{X}_s = u(t)$

----- 1

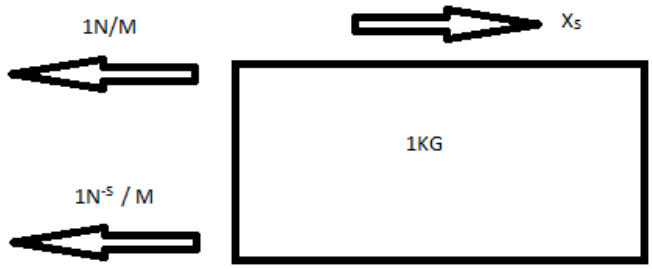


Fig. 3. FBD of mass 2

$-(X_s - X_r) - (\dot{X}_s - \dot{X}_r) = \ddot{X}_s$

Second equations for non-contact of 2nd mass is given above. No force is acting to this mass. One mass and one damper is attached to this system. And the value of coefficients is 1.

$-X_s + X_r - \dot{X}_s + \dot{X}_r = \dot{X}_s$

$\ddot{X}_s + X_s - X_r + \dot{X}_s - \dot{X}_r = 0$ ------2

Taking Laplace gives

$s^2X_r(S) + 2X_r(S) - X_S(S) + 2sX_r(S) - sX_S(S) = u(S)$

$S^2X_s(S) + X_R(S) + SX_S(S) - SX_R(S) = 0$

Solve by using matrix

$X_S(S) / u(S) = (S+1) / (S^4 + 3S^2 + 4S^2 + 2S + 1)$

The transfer function for non contact.

2nd degree of freedom.

……

In the first part, equations were converted to Laplace domain and then transfer function was calculated. Now, equations can be converted to state space. As both equations are 2nd order so 4 state variables can be defined

$X_1 = X_r$, $X_2 = \dot{X}_r$ $X_3 = X_S$ $X_4 = \dot{X}_S$
$\dot{X}_1 = \dot{X}_r = X_2$
$\dot{X}_2 = \ddot{X}_r = -2X_1 - 2X_2 + X_3 + X_4 + U(t)$
$\dot{X}_3 = \dot{X}_S = X_4$
$\dot{X}_4 = \ddot{X}_S = X_1 + X_2 - X_3 - X_4$

Assuming the output to be X_s

$$\dot{X} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -2 & -2 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} U(T)$$

$$Y = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}$$

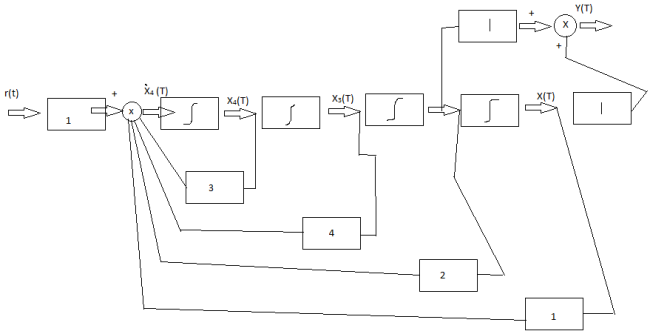
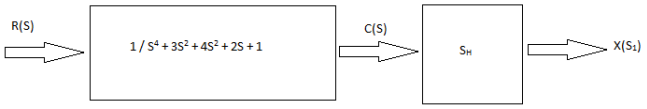


Fig. 4. Block diagram in state space

Block diagram in state space is shown in figure in which input is r(t) and output is u(t)
Firstly, transfer function was converted into state space. For this, firstly converting transfer function Laplace domain into time domain and then 4 variables can be assumed.
From transfer function to state space



$C(S) / R(S) = 1 / (S^4 + 3S^2 + 4S^2 + 1)$
$\dddot{C} + 3\ddot{C} + 4\ddot{C} + 2\dot{C} + C = R$
$X_1 = C$
$\dot{X}_1 = \dot{C} = X_2$
$\dot{X}_2 = X_2 = C$
$X_2 = X_3 = \ddot{C}$
$\ddot{X}_3 = X_4 = \dddot{C}$
$\dot{X}_4 = \dddot{C}$

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & -2 & -4 & -3 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} R(T)$$

$C(S) / X_1(S) = S+1$
$C = \dot{X}_1 + X_1$
$X_2 = \dot{X}_1$

$$Y = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}$$

Now, the state space can be converted into transfer function by using MATLAB. Otherwise, it is difficult to find the inverse od 4*4 matrix.

```
A =

s/(s^4 + 3*s^3 + 4*s^2 + 2*s + 1) + 1/(s^4 + 3*s^3 + 4*s^2 + 2*s + 1)

fx >>
```

By assumptions, the transfer function is
$G(s)=1/4s^2+2s+1$
Natural frequency= 0.5
Damping frequency= 2
For step input

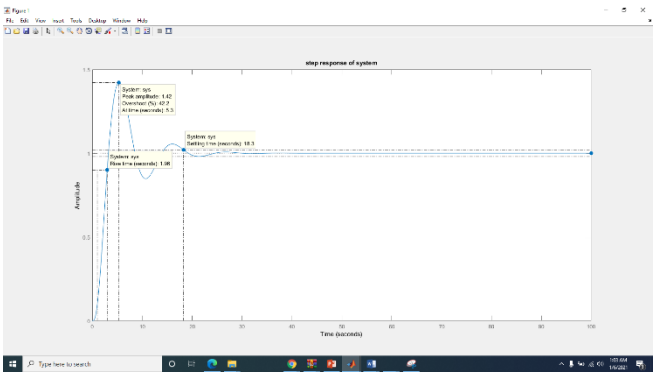


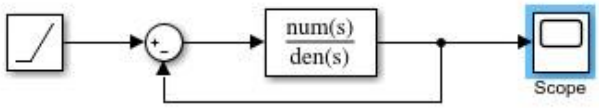
Fig. 5. Step response

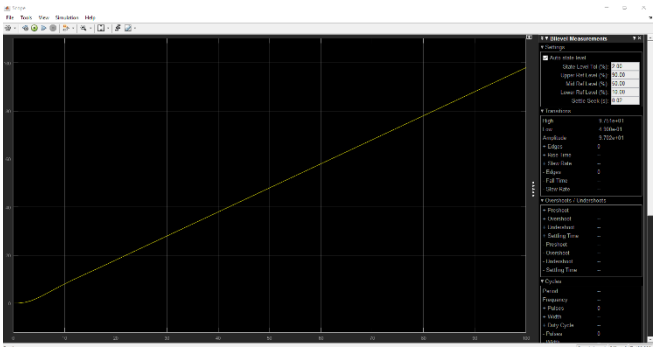
When the step input is given to system, the system shows some oscillations and after some time the oscillations are died out. This means that the system is underdamped. In the first peak the system also shows some overshoot.

TABLE 1 VARIOUS RESPONSES

Settling time	18.3sec
Peak time	5.3sec
Percentage overshoot	42.2percent
Rise time	1.96sec

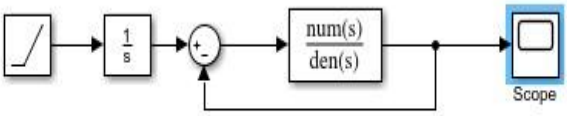
Ramp input

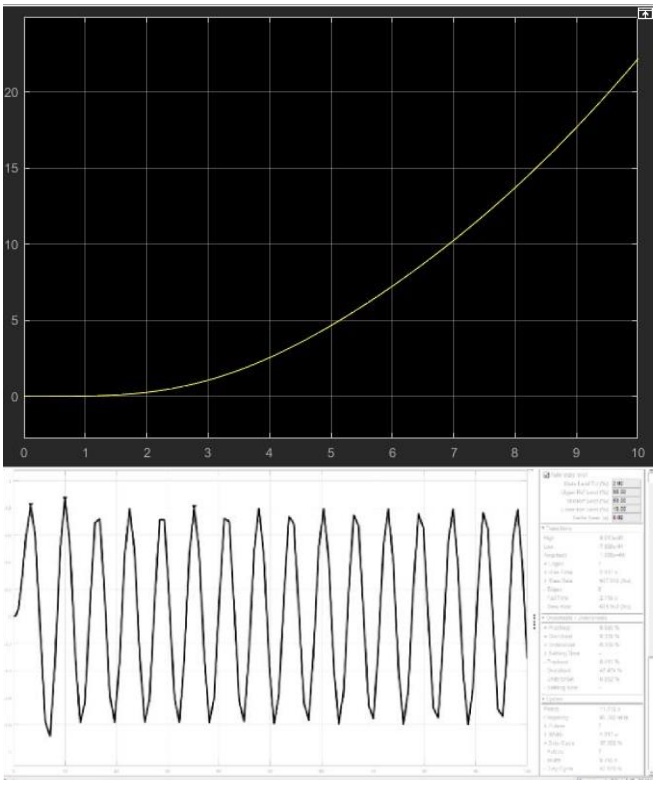




When a ramp input is given to over system the output is constantly increasing as the input is increasing. Because the ramp input is such that the input is constantly increasing.

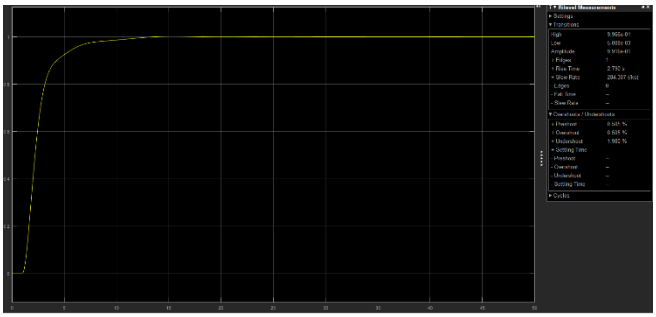
Parabolic respoce:





This is the response of the system for parabolic input. As the parabolic input has oscillations so output has also oscillations.

After adjusting the various responses by adding the PID controller the step response is



Values for PID.

k_p=0.6

k_d=1.5

k_i=0.6

Steady sate error after correction= 1.88%

Rise Time= 2.7sec

Overshoot= 0.5%

Equivalent transfer function

The transfer function can be found by two methods one is reduction of block diagrams and the other one is through masons gain formula. The final equivalent transfer function is same through both methods.

Y(s)=($1/s4 + 3s3 + 4s2 + 2s + 1)(kp + ki/s + skd)$E(s)

E(s)=R(s)-Y(s)

Y(s)=($1/s4 + 3s3 + 4s2 + 2s + 1)(kp + ki/s + skd$)(R(s)-Y(s))

Y(s)(1+ $((1/s4 + 3s3 + 4s2 + 2s + 1kp + ki/s + skd$)= ($1/s4 + 3s3 + 4s2 + 2s + 1)(kp + ki/s + skd)$(R(s))

$$Y(s)/R(s) = \frac{(1)(1/s4 + 3s3 + 4s2 + 2s + 1)(kp + ki/s + skd)}{(1 + (1/(s4 + 3s3 + 4s2 + 2s + 1))\left(kp + \frac{ki}{s} + skd\right))}$$

$$G(s) = \frac{C(s)}{R(s)} = \frac{\sum_k T_k \Delta_k}{\Delta}$$

Identifying forward path gain

$1 * T_1(s) * G(s)$

Loop Gains

$T_1(s) * G(s) * (-1)$

There are no non-touching loop gains.

$\Delta = 1 - (T_1(s) * G(s) * (-1))$

$\Delta = 1 + T_1(s)G(s)$

$\Delta_k = 1$

$$G(s) = \frac{\sum_k T_k \Delta_k}{\Delta}$$

$$G(s) = \frac{T_1(s)G(s)(1)}{1+T_1(s)G(s)}$$

Putting values of $T_1(s)$ and G(s)

$$G(s)= \frac{(1)(1/s4+3s3+4s2+2s+1)(kp+ki/s+skd)}{(1+(1/(s4+3s3+4s2+2s+1))(kp+\frac{ki}{s}+skd))}$$

For the stability of first transfer function the Routh table is shown in table 3.

TABLE 2 ROUTH TABLE

s^4	1	4	1
s^3	3	2	0
s^2	3.33	1	0
s^1	1.1	0	0
s^0	1	0	0

In order to find the steady state error the step input $(\frac{1}{s})$ is applied to the system. The error calculated is

$$e(\infty) = e_{step}(\infty) = \lim_{s\to 0} \frac{s(\frac{1}{s})}{1+G(s)} = \frac{1}{1+\lim_{s\to 0} G(s)} = \frac{1}{1+\lim_{s\to 0} (1/s4+3s3+4s2+2s+1)} = \frac{1}{1+1} = 0.5$$

For the ramp input $(\frac{1}{s^2})$ the steady state error will be

$$e(\infty) = e_{ramp}(\infty) = \lim_{s\to 0} \frac{s(\frac{1}{s^2})}{1+G(s)} = \frac{1}{\lim_{s\to 0} sG(s)} = \frac{1}{\lim_{s\to 0} s(1/s4+3s3+4s2+2s+1)} = \frac{1}{0} = \infty$$

For the parabolic input $(\frac{1}{s^3})$ the steady state error will be

$$e(\infty) = e_{parabolic}(\infty) = \lim_{s\to 0} \frac{s(\frac{1}{s^3})}{G(s)} = \frac{1}{\lim_{s\to 0} s^2G(s)} = \frac{1}{\lim_{s\to 0} s^2(1/s4+3s3+4s2+2s+1)} = \frac{1}{0} = \infty$$

B. For contact

Apply Newton's second law

$$F = ma$$
$$\ddot{Xr} + 2\dot{Xr} + 2Xr - Xs - \dot{Xs} = u(t)$$
$$-\dot{Xr} - Xr + \ddot{Xs} + \dot{Xs} + Xs - Z - Xe = 0$$
$$-Xs + \dot{Z} + Z - \dot{Xe} = \ddot{Z}$$
$$-Xs - \dot{Z} + \ddot{Xe} + 2\dot{Xe} + 2Xe = 0$$

These equations are for the masses 1, 2,3 and small assumed mass.

2nd degree of freedom.

Taking Laplace of second equation

$$s2\,Xr + 2sXr + 2Xr - Xs - sXs = u(s)$$
$$-sXr - Xr + s2Xs + sXs + Xs - Z - Xe = 0$$
$$-Xs + sZ + Z - sXe = \ddot{Z}$$
$$-Xs - sZ + s2Xe + 2sXe + 2Xe = 0$$

Results from matlab

$$G(s) = XS/U(S)$$
$$G(s) = (s3 + 2s2 + 4s + 2)/(s6 + 4s5 + 10s4 + 13s3 + 7s2 - 4s - 4)$$

This is transfer function for non-contact in Laplace domain.

8 variables are assumed

1 $X1 = Xr$
2 $X2 = \dot{X}r$
3 $X3 = Xs$
4 $X4 = \dot{X}s$
5 $X5 = Z$
6 $X6 = \dot{Z}$
7 $X7 = Xe$
8 $X8 = \dot{X}e$

Now developed equations for state space

1 $\dot{X}1 = X2$
2 $\dot{X}2 = -2X1 - 2X2 + X3 + X4 + u(t)$
3 $\dot{X}3 = X4$
4 $\dot{X}4 = X1 + X2 - X3 - X4 + X5 + X7$
5 $\dot{X}5 = X6$
6 $\dot{X}6 = X4 + X3 - 2X8 - 2X7$
7 $\dot{X}7 = X8$
8 $\dot{X}8 = X3 + X6 - 2X8 - 2X7$

$$A = \begin{vmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & -2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & -1 & -1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & -2 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & -2 & -2 \end{vmatrix} \begin{vmatrix} X1 \\ X2 \\ X3 \\ X4 \\ X5 \\ X6 \\ X7 \\ X8 \end{vmatrix} + \begin{vmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix} u(t)$$

Assume X_5 to be output

$$Y = \begin{vmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{vmatrix} \begin{vmatrix} X \end{vmatrix}$$

When converting the transfer function to state space, firslty the laplace domain is converted into time domain.

$$\begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \dot{X}_3 \\ \dot{X}_4 \\ \dot{X}_5 \\ \dot{X}_6 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 4 & 4 & 7 & -13 & -10 & -4 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} R(T)$$

From state space to

$$Y = \begin{bmatrix} 1 & 2 & 4 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{bmatrix}$$

When converting the state space to transfer function, the transfer function was used.

```
X =

[  s,    -1,  0,     0,  0,  0,  0,     0]
[  2, s + 2, -1,    -1,  0,  0,  0,     0]
[  0,     0,  s,    -1,  0,  0,  0,     0]
[ -1,    -1,  1, s + 1, -1,  0, -1,     0]
[  0,     0,  0,     0,  s, -1,  0,     0]
[  0,     0, -1,    -1,  0,  s,  2,     2]
[  0,     0,  0,     0,  0,  0,  s,    -1]
[  0,     0, -1,     0,  0, -1,  2, s + 2]


A =

(s^3 + 2*s^2 + 4*s + 2)/(s^6 + 4*s^5 + 10*s^4 + 13*s^3 + 7*s^2 - 4*s - 4)

fx >>
```

By assumptions the transfer function is

$G(s)=4/7s^2+4s+4$

Natural frequency=0.755

Damping frequency= 0.377

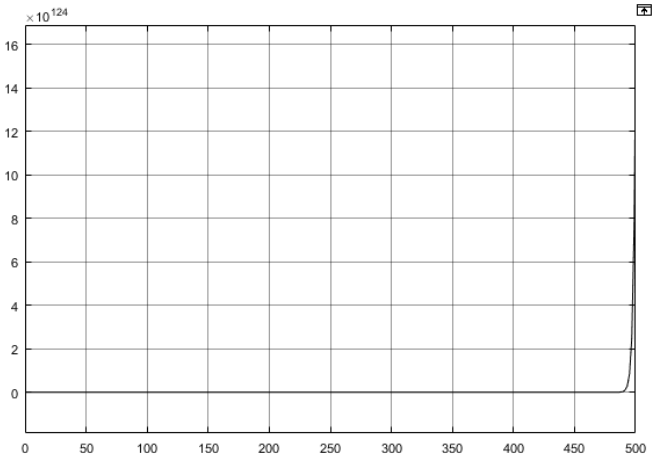


Fig. 6. Step response of system

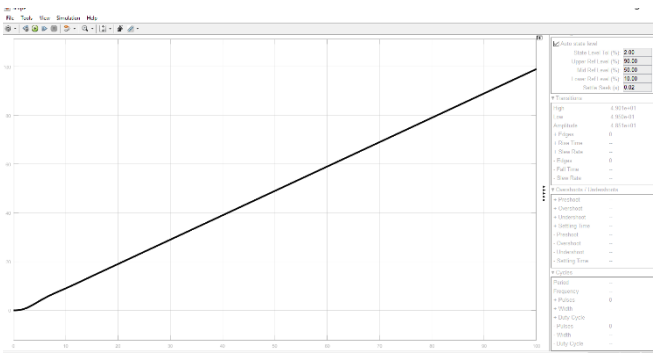


Fig. 7. Ramp input

This is unstable transfer function. To make it stable we need to make changes in modeling and system design.

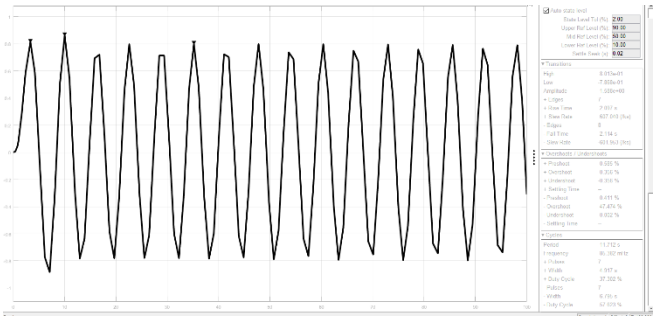


Fig. 8. parabolic input

Equivalent transfer function

$Y(s)=(\quad (s3 + 2s2 + 4s + 2/(s6 + 4s5 + 10s4 + 13s3 + 7s2 - 4s - 4)(kp + ki/s + skd)E(s)$

$E(s)=R(s)-Y(s)$

$Y(s)=(\quad (s3 + 2s2 + 4s + 2/(s6 + 4s5 + 10s4 + 13s3 + 7s2 - 4s - 4)(kp + ki/s + skd)(R(s)-Y(s))$

$Y(s)(1 + (s3 + 2s2 + 4s + 2/(s6 + 4s5 + 10s4 + 13s3 + 7s2 - 4s - 4)(kp + ki/s + skd) \quad) \quad (s3 + 2s2 + 4s + 2/(s6 + 4s5 + 10s4 + 13s3 + 7s2 - 4s - 4)(kp + ki/s + skd)(R(s))$

$$\frac{Y(s)}{R(s)} = \frac{(1)\ (s3 + 2s2 + 4s + 2/(s6 + 4s5 + 10s4 + 13s4 + 7s2 - 4s - 4)(kp + ki/s + skd)}{(1 + (s3 + 2s2 + 4s + 2/(s6 + 4s5 + 10s4 + 13s4 + 7s2 - 4s - 4)(kp + ki/s + skd)}$$

Mason Gain Formula

For solving the mason's rule, firstly find the number of forwards path gain. Secondly. find the loop gains. After this, find the non-touching loops and touching loops.

$G(s) = \frac{C(s)}{R(s)} = \frac{\sum_k T_k \Delta_k}{\Delta}$

Identifying forward path gain

$1 * T_1(s) * G(s)$

Loop Gains

$T_1(s) * G(s) * (-1)$

There are no non-touching loop gains.

$\Delta = 1 - (T_1(s) * G(s) * (-1))$

$\Delta = 1 + T_1(s)G(s)$

$\Delta_k = 1$

$G(s) = \frac{\sum_k T_k \Delta_k}{\Delta}$

$G(s) = \frac{T_1(s)G(s)(1)}{1+T_1(s)G(s)}$

Putting values of $T_1(s)$ and G(s)

$G(s) = \frac{(1)\ (s3+2s2+4s+2/(s6+4s5+10s4+13s4+7s2-4s-4)(kp+ki/s+skd)}{(1+(s3+2s2+4s+2/(s6+4s5+10s4+13s4+7s2-4s-4)(kp+ki/s+skd)}$

Routh table for second transfer function

TABLE 3 ROUTH TABLE

S^6	1	10	7	-4
s^5	4	13	-4	0
s^4	6.75	8	-4	0
s^3	8.25	-1.62	0	0
s^2	9.33	-4	0	0
s^1	1.91	0	0	0
s^0	-4	0	0	0

Steady state error for

$$e(\infty) = e_{step}(\infty) = \lim_{s\to 0} \frac{s\left(\frac{1}{s}\right)}{1+G(s)} = \frac{1}{1+\lim_{s\to 0} G(s)} = \frac{1}{1+\lim_{s\to 0} (s3+2s2+4s+2/(s6+4s5+10s4+13s3+7s2-4s-4)} = \frac{1}{1-0.5} = 2$$

For the ramp input $(\frac{1}{s^2})$ the steady state error will be

$$e(\infty) = e_{ramp}(\infty) = \lim_{s\to 0} \frac{s\left(\frac{1}{s^2}\right)}{1+G(s)} = \frac{1}{\lim_{s\to 0} sG(s)}$$

$$\frac{1}{\lim_{s\to 0} s(s3+2s2+4s+2/(s6+4s5+10s4+13s3+7s2-4s-4)} = \frac{1}{0} = \infty$$

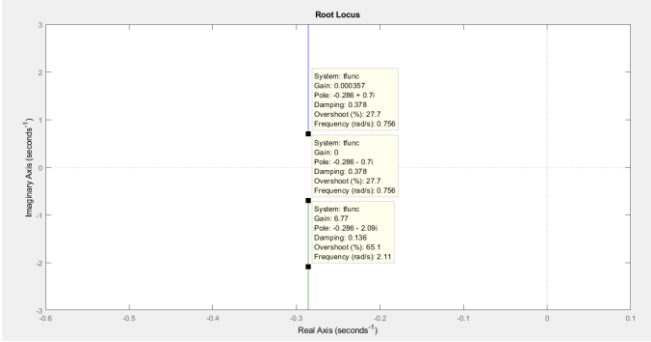
For the parabolic input $(\frac{1}{s^3})$ the steady state error will be

$$e(\infty) = e_{parabolic}(\infty) = \lim_{s\to 0} \frac{s(\frac{1}{s^3})}{G(s)} = \frac{1}{\lim_{s\to 0} s^2G(s)}$$

$$\frac{1}{\lim_{s\to 0} s^2(s3+2s2+4s+2/(s6+4s5+10s4+13s3+7s2-4s-4)} = \frac{1}{0} = \infty$$

```
clc
clear all
num= [4]
den= [7 4 4]
tfunc=tf(num,den);
rlocus(tfunc)
```

Root l;ocus of our transfer function is



III. CONCLUSION

As one of transfer function is stable because when plotting the Routh table for the transfer function there is no sign changes so there is no poles in right half plane, so system is stable while finding poles by using MATLAB. As all values has negative sign so it lies on left half plane. But for the second transfer function when plotting the Routh table there is two sign changes so the two poles lies in right half plane so initially this transfer function is unstable. For first transfer function, the steady state error was calculated for step input which is o.5% after correction and for ramp and parabolic input the error is infinity. And for second transfer function the steady state error for step input is two and for ramp and parabolic input error is infinity. The steady state error for step input for second function is higher. When we applied PID controller to second transfer function it remains unstable.

Robotic manipulators typically find applications in the following:

A. *Handling of radioactive and biohazardous materials*

B. *Robot-assisted medical surgeries*

C. *Ocean engineering systems*

IV. RECOMMENDATIONS

Experiment the task-based decision making and control framework on physical robots Develop tool/task models for several common robotic tasks for better understanding of tool and task relations. Extend the parameter learning algorithm to other subjective parameters such as perturbation size and momentum term. Comprehensive literature review on machine learning and other artificial intelligence techniques and their applications to advanced robotics, especially for skill acquisition. Develop a high-performance force/motion control method or perhaps a collection of methods suitable for various robotic tasks for integration with TBRR. Investigate the effects of system parameters on performance reserves and how they can be used in condition-based maintenance and fault tolerance. One major issue of the parallel control approach is that it allows to account geometric and dynamic characteristics of the environment at different levels in the system design. This feature is appealing as it gives proper roles to each component of the control system. It seems to be effective, indeed, to control the dynamics of the interaction while letting the geometry of the

task be accounted at the planning level. An impact is mostly due, indeed, to (unavoidable) planning errors and then it cannot be handled by any strategy that uses the same information on which the task planning is based. Accordingly, no information about the location and orientation of the environment is required in the design of a parallel controller.

V. REFERENCES

Zengxi Pan Hui Zhang, (2008), "Robotic machining from programming to process control: a complete solution by force control" Industrial Robot: An International Journal, Vol. 35 Iss 5 pp. 400 – 409

[1] Zengxi Pan Hui Zhang, “robotic machining from programming to process control: a complete solution by force control” Industrial Robot: An International Journal, Vol.35 Iss 5 pp.(2008)

[2] G. Zhang and J. Furusho, "Control of robot arms using joint torque sensors", IEEE Control Syst. Mag., vol. 18, no. 1, pp. 48-55, 1998.

[3] Zhi-Ren Tsai, Jiing-Dong Hwang, "$H\infty$ Neural-Network-Based Discrete-Time Fuzzy Control of Continuous-Time Nonlinear Systems with Dither", Mathematical Problems in Engineering, vol. 2012, Article ID 314964, 18 pages, 2012

[4] Li-lian Huang, Jin Chen, "Fuzzy PD Control of Networked Control Systems Based on CMAC Neural Network", Mathematical Problems in Engineering, vol. 2012

[5] Qing-Quan Liu, Fang Jin, "LQG Control of Networked Control Systems with Limited Information", Mathematical Problems in Engineering, vol. 2014, Article ID 206391, 12 pages, 2014

[6] Jie Jiang, Changlin Ma, "Comprehensive Control of Networked Control Systems with Multistep Delay", The Scientific World Journal, vol. 2014, Article ID 814245, 8 pages, 2014

[7] hanbin Li, Dominique Sauter, Christophe Aubrun, Joseph Yamé, "Stability Guaranteed Active Fault-Tolerant Control of Networked Control Systems", Journal of Control Science and Engineering, vol. 2008, Article ID 189064, 9 pages, 2008.

[8] Sung Hyun Kim, " Output-Feedback Tracking Control for Networked Control Systems", Mathematical Problems in Engineering, vol. 2015, Article ID 724389, 10 pages, 2015

[9] Hongbo Li, Fuchun Sun, Zengqi Sun, "Delay-Dependent Fuzzy Control of Networked Control Systems and Its Application", Mathematical Problems in Engineering, vol. 2013, Article ID 691370, 9 pages, 2013

[10] Amal Tiab, Louiza Bouallouche-Medjkoune, "Routing in Industrial Wireless Sensor Networks: A Survey", Chinese Journal of Engineering, vol. 2014, Article ID 579874, 7 pages, 2014

[11] P. Antsaklis and J. Baillieul, “Special issue on technology of networked control systems,” Proceedings of the IEEE, vol. 95, no. 1, pp. 5–8, 2007.

[12] M. S. Mahmoud and M. Sabih, “Experimental investigations for distributed networked control systems,” IEEE Systems Journal, vol. 8, no. 3, pp. 717–725, 2014.

[13] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators", Trans. ASME, vol. 102, pp. 126-133, June 1981.

[14] Modern Mechanical Engineering Vol. 3 No. 2A (2013) , Article ID: 33474 , 8 pages Application and Analysis of Force Control Strategies to dubirring and grinding. Frank Domroes, Carsten krewet, Bernd kuhlenkoetter . Institute of production system, TU Dortmund University, Dortmund, Germany.

[15] Xiaohu XU, Dahu ZHU, Haiyang ZHANG, Sijie YAN, Han DING. Application of novel force control strategies to enhance robotic abrasive belt grinding quality of aero-engine blades.

[16] A novel force and motion control strategy for robotic chamfering of gears. Jie Hu, Prabhakar R. Pagilla. *Texas A&M University, College Station, TX 77840 USA*

[17] Fuzzy-Sliding Mode Force Control Research on Robotic Machining. Shou-yan Chen, Tie Zhang, and Yan-biao Zou. *School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou, Guangdong 510640, China* . Correspondence should be addressed to Tie Zhang; merobot@scut.edu.cn . Received 10 January 2017; Revised 18 April 2017; Accepted 23 April 2017; Published 18 May 2017

[18] ABB Robot User Manual (1981), ABB Robot User Manual, IRC5 Reference. Basanez, L. and Rosell, J. (2005), “Robotic polishing systems – from graphical task specifification to automatic programming”, IEEE Robotics & Automation Magazine, Vol. 12, pp. 35-43

[19] Y. Chen and F. Dong, “Robot machining: recent development and future research issues,” *International Journal of Advanced Manufacturing Technology*, vol. 66, no. 9–12, pp. 1489–1497, 2013.

[20] Coope, I.D. (1993). Circle fifitting by linear and nonlinear least squares. Journal of Optimization Theory and Applications, 76(2), 381–388.

[21] https://www.hindawi.com/journals/complexity/2020/3156787/