Application of Linear MPC-CTC Hybrid Controller to IRp_6 Robot

Ali Benniran

Information & Automation Department,
Warsaw University of Technology
Warsaw, Poland, E-mail: benniran 1@yahoo.co.uk

الملخص

تعرض هذه الورقة دراسة تطبيق النظام الخطي التقليدي نوع PD_CTC و النظام الخطي التنبؤي MPC على رجل آلي ذو 6 درجات من الحرية. النظام التقليدي PD_CTC بسيط ويعامل كل ذراع من اذرع رجل آلي على أنها حلقة تحكم منفصلة، بينما النظام التنبؤي MPC يتعامل معها كوحدة متكاملة، علاوة على ذلك فإن للرجل الآلي قيود فنية يجب عدم تجاوزها. النظام التقليدي غير مهيأ للتعامل معها على عكس النظام التنبؤي الذي يأخذها بعين الاعتبار أثناء عملية السيطرة. تم استعمال نموذج بيتر كوك Peter Coke Model لغرض حساب معاملات القصور والاحتكاك والجاذبية، كما تم تحديد مسارات متغيرات المفاصل ثلاثية الدرجة واعتبارها المسارات المرغوب إتباعها لكل مفصل. تم تطوير نظام خطي هجين يتمتع بميزات كلا النظامين المذكورين أعلاه. و أظهرت محاكاة النظام الهجين مزايا جديرة بالاهتمام.

ABSTRACT

This paper studies the application of PD-linear Computed Torque Control (CTC-PD) and linear Model-based Predictive Control (MPC) techniques to IRp-6 robot with six degrees of freedom. The CTC-PD technique is simple but it treats each joint of the robot as separate control loop (SISO) system while the MPC technique is computationally more involved, however it considers the robot as one unit (MIMO) system. Moreover the robot has stringent constraints on inputs and outputs which further complicate the task for the CTC-PD controller. Peter Corke robot model functions' Toolboxes were used to calculate the robot inertia matrix and the gravity, friction and centrifugal force vectors. The desired joints cubic trajectories values are calculated and used for positions and velocities set-point trajectories. A hybrid linear MPC-CTC-PD controller was developed. This scheme contains the advantages of both separate controllers. A simulation program was prepared using the MATLAB, the results of simulation shows benefits of the hybrid solution.

KEYWORDS: IRp-6 robot; Linear MPC control; CTC-PD control; hybrid control system.

INTRODUCTION

The applications of the robotics in industry and other fields are in progress increasingly [1,2]. This requires high quality and precision performance. Simple PDs, PIDs were the dominant controllers up to the latest of the last century. They achieve a satisfactory performance, however, they are single-input single-output (SISO) and unable to handle process constraints [3]. The CTC-PD is an enhanced PD controller. It involves the process model. For CTC-PD to be high performance technique, the process model should be accurate enough. Robots are multi-variable processes characterized by their nonlinearity and strong interaction between their links (joints). Moreover, they have model uncertainties due to payload and friction changes and specified workspaces. These complicated the tasks for CTC-PD controller and at the same time excited the theorists to search for more advanced control. Consequently the advanced controllers such as linear and non-linear Model-based Predictive Control (MPC) get more attention from theorists and practitioners. In the last two decades many successful application of MPC to different areas including the robots appeared [4-9].

MPC has the advantage of involving the multi-in multi-out (MIMO) explicit process model and capability to handle different process constrains (on inputs, outputs…). The process model used in linear MPC technique is the linear process model, while in non-linear MPC technique non-linear model is used. The main disadvantage of the MPC technique is its large amount of on-line computations. However the linear MPC optimizing a quadratic programming problem (QP) is an efficient computation technique, the solution algorithm converges reliably to the optimum in short time [10-12].

The linear MPC technique is well established, it provides the majority of the possible benefits with MPC technology [4]. In [7] the linear predictive functional control (PFC) was successfully applied to KUKA robot, in [8, 9] a linearized-PFC was applied to SCARA and PUMA-560 robots, whereas in [13, 14] a nonlinear successive Linearization MPC-NSL technique was applied to IMI and IRp-6 robot, they prove advantageous regarding robustness and disturbance rejection.

This paper presents an application of both linear CTC-PD and linear MPC controllers to a 6-DOF, IRp-6 robot. The IRp-6 linear model is a result of using the robot inertia matrix, Coriolis and centrifugal, friction and gravity vectors which were computed via application of Peter Corke robot model relevant functions Toolboxes for positions at zeros [15]. The joint space trajectories (set-point) toolbox of 7th order polynomial was used for generating the desired trajectories. The paper presents a hybrid scheme gathering the advantages of the linear CTC-PD and linear MPC techniques. The simulation of the hybrid scheme shows interesting results.

MANIPULATOR DYNAMIC MODEL

The general form of the rigid body manipulator dynamic equations, see Figure. (1), known as Euler Lagrange's equation is:

$$\tau = M(q)\ddot{q} + N(q,\dot{q}) + G(q) + F(q,\dot{q}) \tag{1}$$

where; $\tau \in \Re^n$ is the inputs torque vector, $M(q) \in \Re^{n\times n}$ is the symmetrical positive definite manipulator inertia matrix, $N(q,\dot{q}) \in \Re^n$ is the centrifugal and Coriolis terms vector, $G(q) \in \Re^n$ is the gravity vector and $F(q,\dot{q}) \in \Re^n$ is the viscous and Coulombs friction vector, $q \in \Re^n$ is the generalized joints position vector, $\dot{q} \in \Re^n$ is the joints velocity vector and n is the number of joints also equal to the number of DOF.

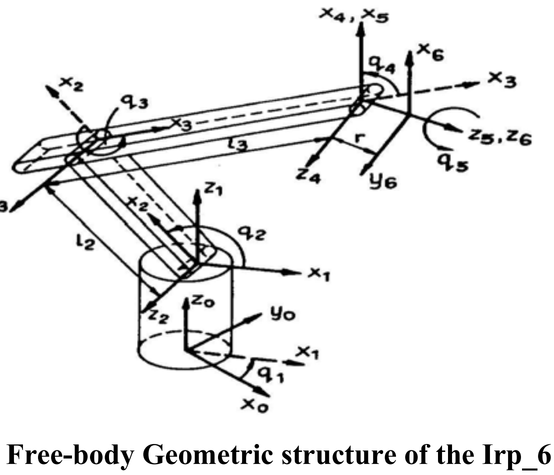

Figure 1: Free-body Geometric structure of the Irp_6 Robot

There are two methods for formulating equation (1). 1- Euler-Lagrange method. 2- The recursive Newton-Euler method. Both methods require detailed robot parameters, such as link masses, first and second inertia and their mass centres, motors inertia and gears ratio. The two approaches lead to the same formula [16, 17, 3]. Table (1) presents IRp_6 links parameters based on m. D-H*, these parameters are usually given by the manipulator's manufacturer.

The IRp-6 links mass and inertia values are rarely revealed by the manipulator's manufacturer. Furthermore, the experimental measurements of these values are very complicated; it involves robot dismantling and identification [3, 16, 17]. However, in [18, 19] valuable data regarding the masses, link lengths, mass centres and inertia are given, but the distribution of links inertia tensors were missing.

Table 1: Links parameters of the IRp_6 Robot using modified D-H convention

i	a_{i-1}	α_{i-1}	d_i	ϑ_i	θ_I limits
1	0	0	0	ϑ_1	$-170<\theta_1<170$
2	0	$-\pi/2$	0	ϑ_2	$-150<\theta_2<50$
3	0.45	0	0	ϑ_3	$-25<\theta_3<40$
4	0.67	0	0	ϑ_4	$-90<\theta_4<90$
5	0	$-\pi/2$	0.15	ϑ_5	$-180<\theta_5<180$
6	0	$\pi/2$	0	ϑ_6	$-180<\theta_6<180$

*m. D-H is the modified Denavit-Hartenberg convention

From the mechanical structure of the manipulator, Figure (1), and the general assumption of robot symmetry with inertia principle axes the values shown in Table (2) were evaluated. The data of tables (1 and 2) are used in Peter Corke robot model template for generating the IRp-6 robot object which will be used in this paper. For checking the validity of the founded inertia values, the IRp-6 robot simulated with zero inputs torques for:

- Study the steady state response
- Compare it with the response of PUMA560 which is well established

Table 2: IRp-6 inertias

Link nr	I_{xx}	I_{yy}	I_{zz}	I_{xy}	I_{xz}	I_{yz}	m link mass	J_m motor inertia	η
Link 1	1.57	1.57	1.57	0	0	0	0	1.64e-4	158
Link 2	0.79	0.79	0	0	0	0	8	160e-6	158
Link 3	1.86	7.43	7.43	0	0	0	29.78	160e-6	158
Link 4	160e-6	160e-6	160e-6	0	0	0	1	160e-6	128
Link 5	160e-6	160e-6	160e-6	0	0	0	3	160e-6	128
Link 6	4.096	0	0	0	0	0	1	160e-6	128

The validation checks result (not shown here for lack of space) show validity of the selected inertia tensor values. The steady state response shows that the equilibrium points (EP) are:

$q_1=0$ (arbitrary); $q_2=-\pi(m+1)/2$; $q_3=0$; $q_4=-m\pi$; $q_5=-\pi(m+1)/2$; $q_6=-\pi(m+1)/2$;

$$m = 0, 1, 2\ldots \quad (2)$$

For control purpose, Peter Corke robot model Toolboxes have been used, at each sampling instant, for generating IRp-6 robot matrices and vectors.

DESIRED TRAJECTORIES

The desired (reference) trajectories $q_d = [q_d^1, q_d^2, ...q_d^n]$ are the manipulator joint paths from their initial positions to their goal positions. The desired joint trajectories are computed from application of the Peter Corke trajectory function toolbox jtraj with end-effector initial position q_{int} and final (goal) position q_g corresponding to the position and orientation of the end-effector off and on the workspace in a duration time t_f.

q_{int} = [0, -pi/4, 0, -pi, -pi/2, -pi/2] the manipulator is at rest (off the workspace).

q_g = [pi/4, pi/4, -pi/2, 0, 0, 0] the manipulator is at the goal position (on the workspace).

t_f , depends on the required smoothness of the trajectory.

Constraints

The general formula for computing motor torque is:

$$\tau_{motor} = k_a \times I$$
$$\tau_{jo\text{int}} = \tau_{motor} \times \eta \tag{4}$$

Where, τ_{motor} is the torque exerted by the motor, $\tau_{jo\text{int}}$ is the torque applied to the joint, η is the gear ratio, k_a=0.105 Nm/A is the current-torque constant and I is the motor current.

The technical limitations of the IRb-6 robot motors input currents are given as follows:

- first joint motor $I_{\max}$= 27 A, η=158, from (4): $\tau_{motor} \approx$ 450 Nm for first joint.
- second and third joint motors, $I_{\max}$=15 A, η=158, from (4): $\tau_{motor} \approx$ 250 Nm
- rest of joint motors, $I_{\max}$=5 A, , η=128, from (4): $\tau_{motor} \approx$ 67 Nm

Therefore the joint limit (constraint) input torque values are:

$$\tau_{motor\,(\max/\min)} = \pm[2.85;\ 1.58;\ 1.58;\ 0.52;\ 0.52;\ 0.52]$$
$$\tau_{jo\text{int}(\max/\min)} = \pm[450,\ 250,\ 250,\ 67,\ 67,\ 67] \tag{5}$$

LINEAR CTC-PD CONTROLLER

For calculating the joints position and velocity, the forward manipulator dynamic equation is, at each control sample, numerically solved using modified Euler's method with τ as input torque vector and step size $T_p = 0.01s$, the forward dynamics equation is:

$$\ddot{q} = -M^{-1}(N + F + G - \tau) \tag{6}$$

Calculation of the elements of the inertia matrix M , Coriolis and centripetal N , friction F and gravity G vectors are repeatedly generated from using Peter Corke model relevant function Toolboxes for the current joint positions $q \in \Re^n$ and joint velocities $\dot{q} \in \Re^n$.

The PD Computed Torque Control (CTC-PD), also called Inverse Dynamic Control, is involving the robot parameters in computing the joint input torques. Its control algorithm has the form:

$$\tau = M(q_0)(Kp \cdot e + Kv \cdot \dot{e}) + N(q_0, \dot{q}_0) + F(q_0, \dot{q}_0) + G(q_0) \tag{7}$$

where, $e = q_d - q \in \Re^n$, $\dot{e} = \dot{q}_d - \dot{q} \in \Re^n$ and $q_0, \dot{q}_0 \in \Re^n$ are the zero-joints position and velocity respectively (generally any equilibrium point can be used), K_p and K_v are proportional and derivative gain diagonal matrices of proper dimensions.

RESULTS OF SIMULATION OF IRB-6 UNDER CTC-PD CONTROLLER WITH ACTUATORS INPUT CONSTRAINTS

The criterion for tuning is to get as close as possible position tracking without any overshooting. The trail and error method has been applied. Controller parameters correspondingly are: sampling time T_s=0.01 s, K_p=[1300; 1350;1550; 1100; 1100; 1300], K_v=[27; 24; 25; 22; 24; 25]. The input torque limits vector is (5).

Figures (3a and 3c) show the simulation result of IRp-6 under CTC-PD technique, it is noticed that the controllers achieve high position tracking performance with minimum joints time-wise position error along the trajectories. It is also measured that the execution time is 11, 05s

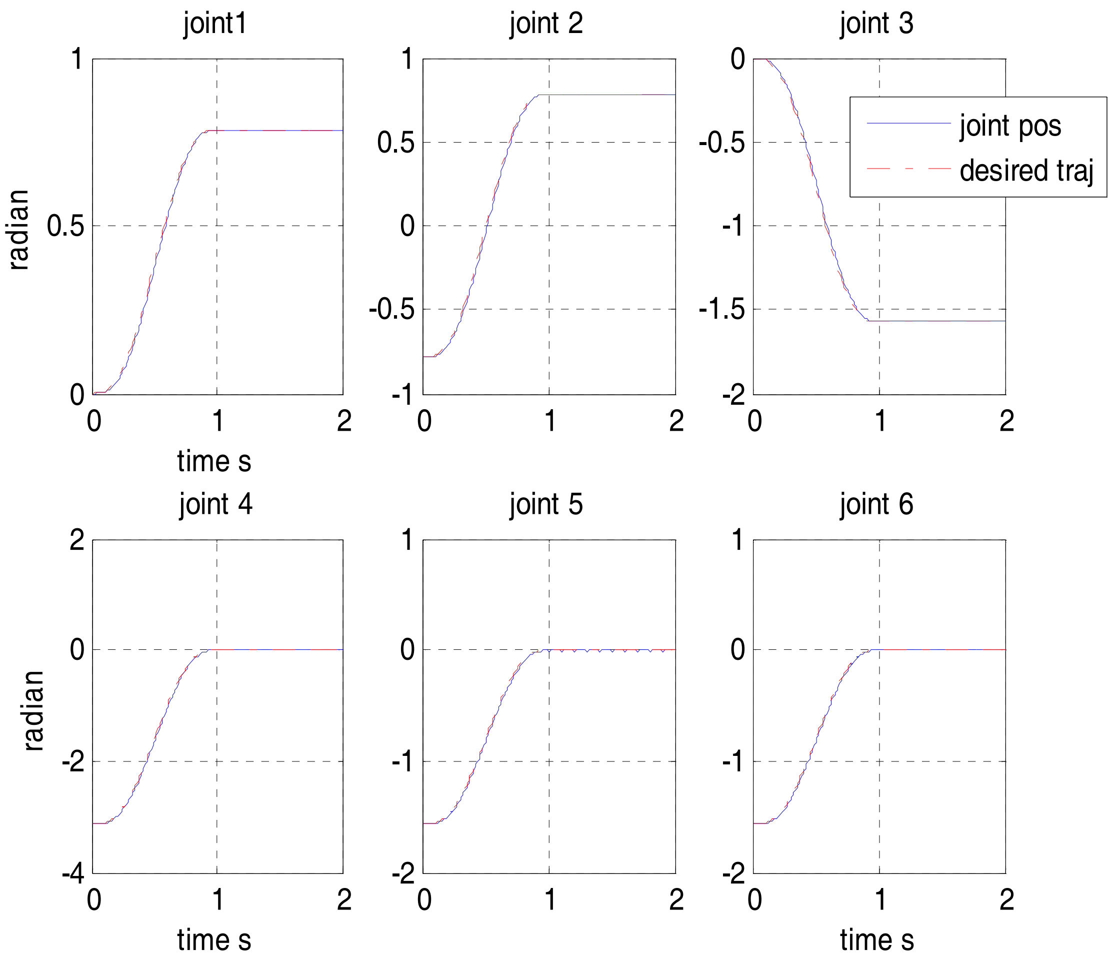

Figure 3a: IRp-6 joint position tracking under constrained linear CTC-PD controllers

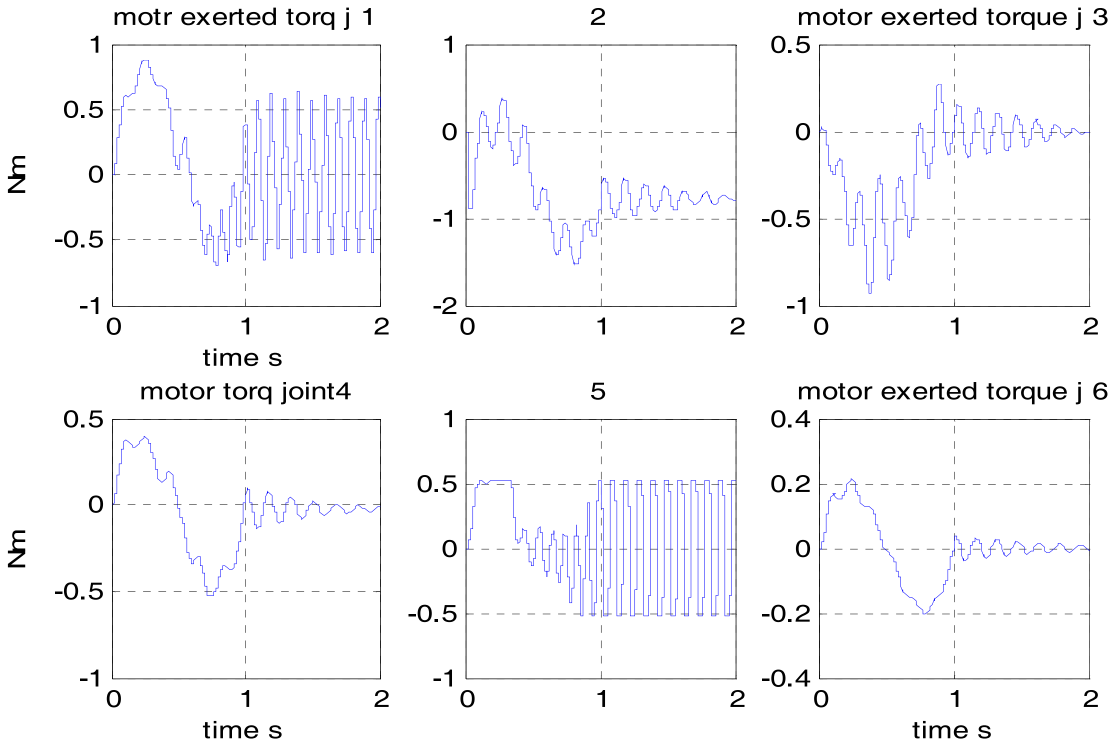

Figure 3b: IRp-6 joint motor torque under constrained linear CTC-PD controllers

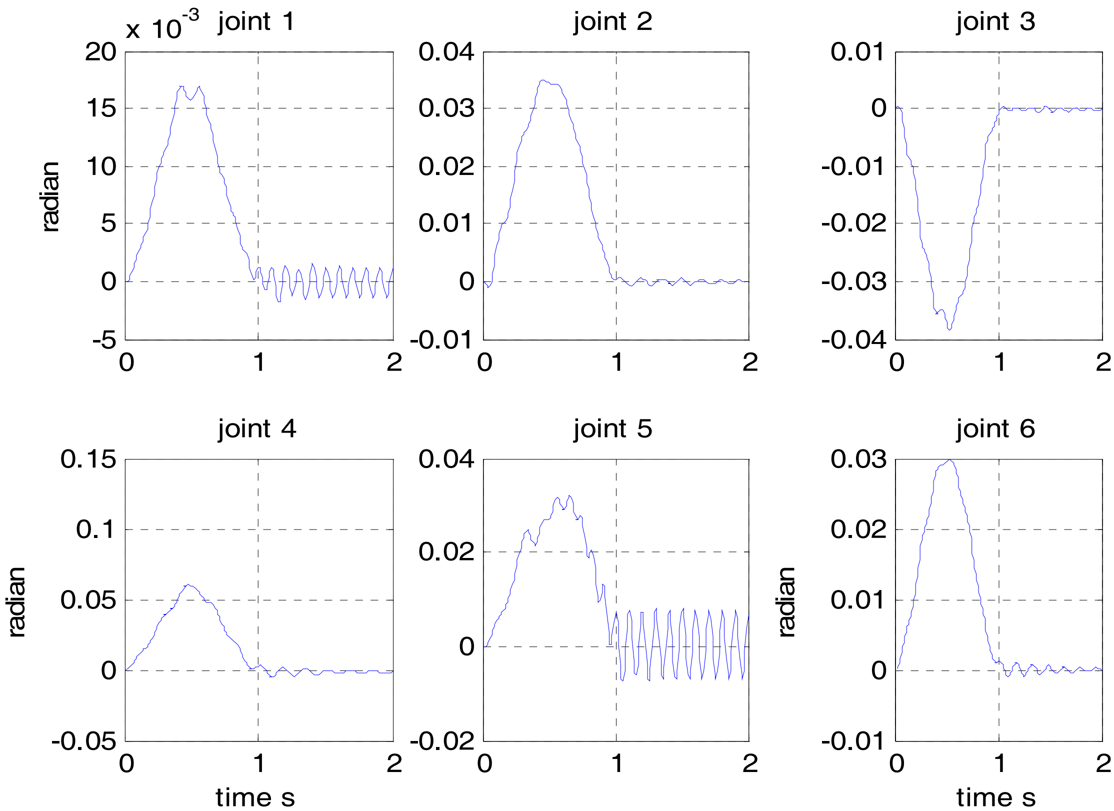

Figure 3c: IRp-6 joints position error under constrained linear CTC-PD controllers

MPC CONTROLLER

The state space model describing a linear system (in continuous time) is given by:

$$\dot{x}(t) = Ax(t) + Bu(t) + v(t)$$
$$y(t) = Cx(t) + d(t) \tag{8}$$

where $x \in \Re^{n_x}$ is the state vector, $u \in \Re^{n_u}$ is the input vector and $v \in \Re^{n_x}$ is the state disturbance vector (measurement and modelling errors), n_y is the number of outputs and n_u is the number of inputs, $n_x = 2n_u$, and $d \in \Re^{n_y}$ is output disturbances.

In robotics the matrices; A, B and C are calculated from the formulae [10]:

$$A = \left[0_{n_u \times n_{uy}} \quad I_{n_u \times n_{uy}}; 0_{n_u \times n_{uy}} \quad diag(M_0^{-1}(N_0 + F_0 + G_0))\right]$$
$$B = \left[0_{n_u \times n_u}; M_0^{-1}\right] \tag{9}$$
$$C = \left[I_{n_y \times n_y}; \ 0_{n_y \times n_y}\right]$$

where, M_0 is the robot positive definite inertia matrix, N_0 is a vector of the centrifugal and Coriolis terms, G_0 is a vector of the gravity terms and F_0 is a vector of the friction terms calculated at zero (generally at any equilibrium point) position and velocity and the state vector is $x = [q^T, \dot{q}^T]^T \in \Re^{n_x}$ where $q, \dot{q} \in \Re^{n_y}$ are the joint position and velocity respectively.

The c2d Matlab function and 'ZOH' discretization method, with sampling time $T_s \in [1ms - 20ms]$ was used to change (8) to discrete time.

The principle of MPC algorithm is to compute at each sampling instant k a set of future optimal control increments ΔU on control horizon Nu, [10]:

$$\Delta U(k) = [\Delta u(k/k)^T, \Delta u(k+1/k)^T \ldots\ldots \Delta u(k+Nu-1/k)^T]^T \tag{10}$$

$$\Delta u(k) = [\Delta u_1(k/k), \Delta u_2(k/k) \ldots\ldots \Delta u_{n_u}(k/k)]^T$$

Where, $\Delta u(k+p/k) = 0 \ \forall p \geq Nu$. The individual i^{th} control input is:

$u_i(k) = \Delta u_i(k) + u_i(k-1)$, $i = 1, 2 \ldots n_u$ and then:

$$u(k) = [u_1(k), \ldots, u_{n_u}(k)]^T \tag{11}$$

The optimal increments vector ΔU is a result of minimization of standard MPC optimization problem [10]:

$$\min_{\Delta U(k)} \{J_{MPC}(k) = \sum_{p=1}^{N_p} \|q_d(k+p/k) - y(k+p/k)\|_\psi^2 + \sum_{p=0}^{N_u} \|\Delta u(k+p/k)\|_\lambda^2\} \tag{12}$$

Subject to:

$$U_{\min} \leq U(k-1) + J\Delta U(k) \leq U_{\max}$$
$$-\Delta U_{\max} \leq \Delta U(k) \leq \Delta U_{\max}$$
$$q_{\min} \leq Y^{prd}(k) \leq q_{\max}$$

where, $q_d(.)$ is the desired trajectory (set-point) vector of length n_y, $y(.)$ is the predicted output vector of length n_y, N_p the prediction horizon and ψ is a positive definite diagonal weighting matrix of dimension $n_y \times n_y$ whereas λ is a semi-definite diagonal weighting matrix of dimension $n_u \times n_u$. J is $n_u.N_u \times n_u.N_u$ matrix its upper diagonal elements are zeros and lower are identity of $n_u \times n_u$, $q_{\min}, q_{\max} \in \Re^{n_y.N_p}$ are the admissible maximum and minimum values of the joint positions (Table 1), while $\Delta U_{\max}, U_{\min}, U_{\max} \in \Re^{n_u.N_u}$ are max and min the input constraints (5).

In MPC only the first set of the input vector, corresponding to the sampling instant *k*, is applied to the process. The process is repeated each successive sample with the same prediction horizon length N_p, but shifted one sample forward. The predicted output is:

$$Y^{prd}(k) = [y(k+1/k)^T, ...y(k+N_p/k)^T]^T \quad (13)$$

The controller is tuned by selecting appropriate values for ψ and λ. Furthermore, applying the same criterion used in classical technique for better position tracking free from overshooting, the desired trajectories are may smoothed by applying the principle of reference trajectory in place of desired trajectory [10, 11]; $q^{ref}(k+p/k) = q_d(k+p) - \gamma(q_d(k) - q(k))$
where $\gamma \in (0, 1)$.

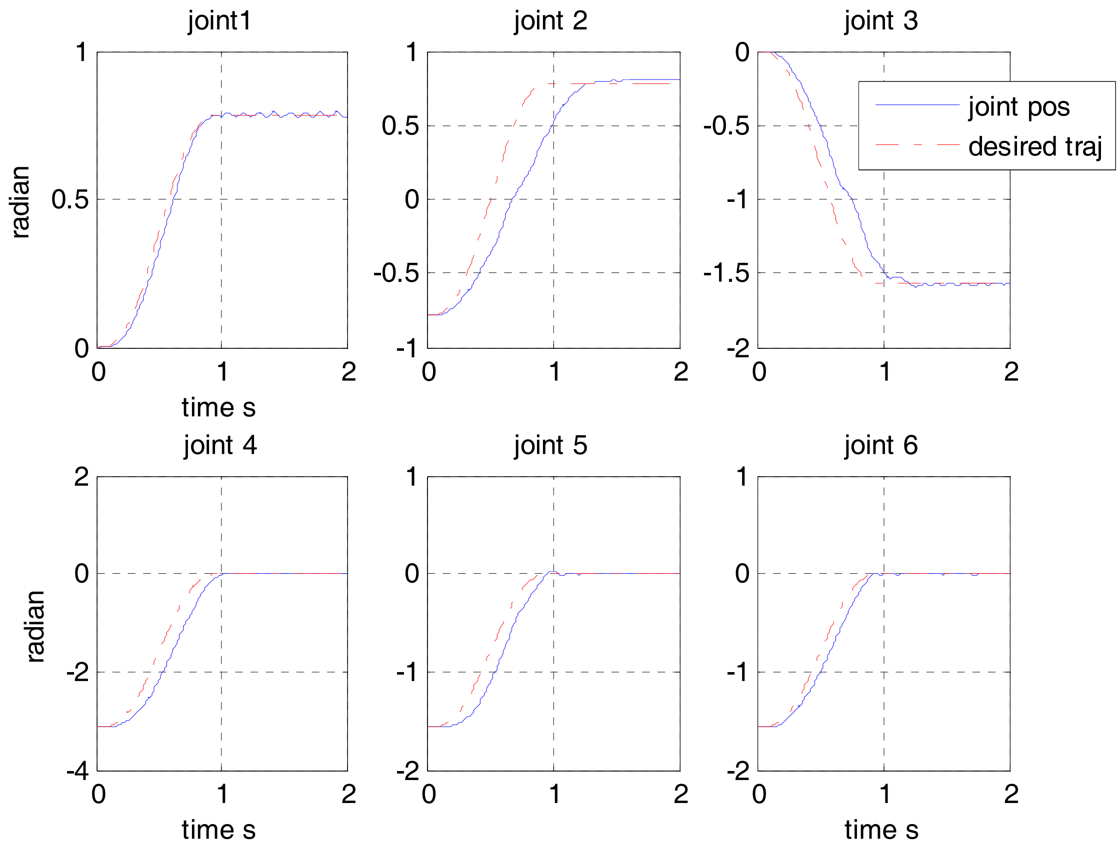

Figure 4a: joint position tracking under constrained linear MPC technique

RESULTS OF SIMULATION OF IRP-6 UNDER CONSTRAINED MPC CONTROLLER

Control parameters: N_p=40, N_u=6, sampling time T_s= 0.01 s. Application of trail and error method leads to the following tuning parameters:
λ = [1E-6; 10E-6; 30E-6; 200E-6; 100E-6; 100E-6], ψ = [600; 600; 700; 1000; 1000; 2000], t_f =1s. The input constraints vector is (5), whereas output constraints are those tabulated in Table (1).

Figures (4a-c) show the IRp-6 joints position tracking and time-wise position errors under input and output constrained linear MPC technique for trajectories of t_f =1s, the controller is tuned only with the penalty weight parameters. However, more smooth joints output trajectories are achieved when the desired trajectories are replaced by reference trajectories with time constant of 1 s. It is worth to notice; from comparing of figure (3b) and figure (4b), that the required inputs torque are higher under CTC-PD technique than under MPC technique. This is expected because of optimization process in the later technique. It is measured that the execution time is 16, 50s.

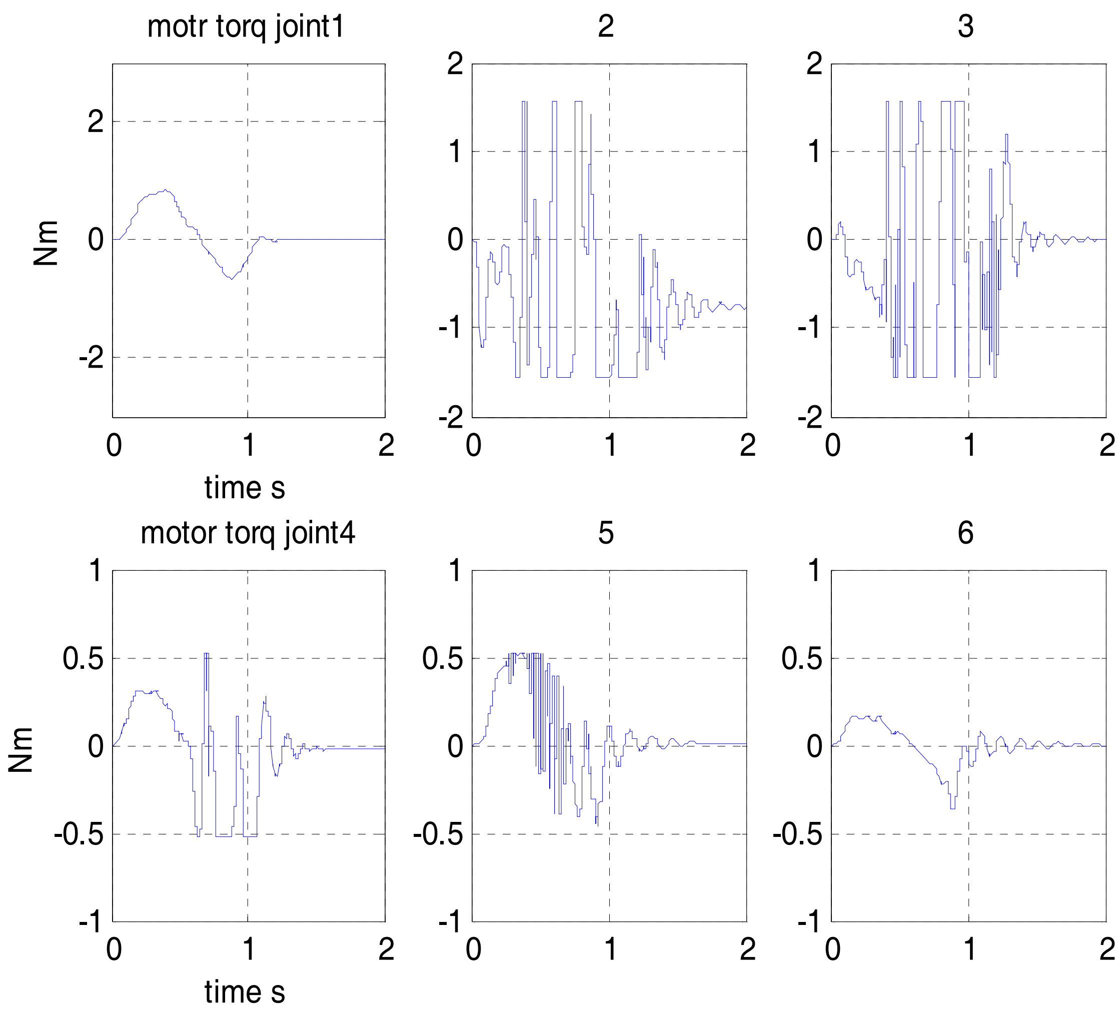

Figure 4b: joint motor exerted torque under constrained linear MPC technique

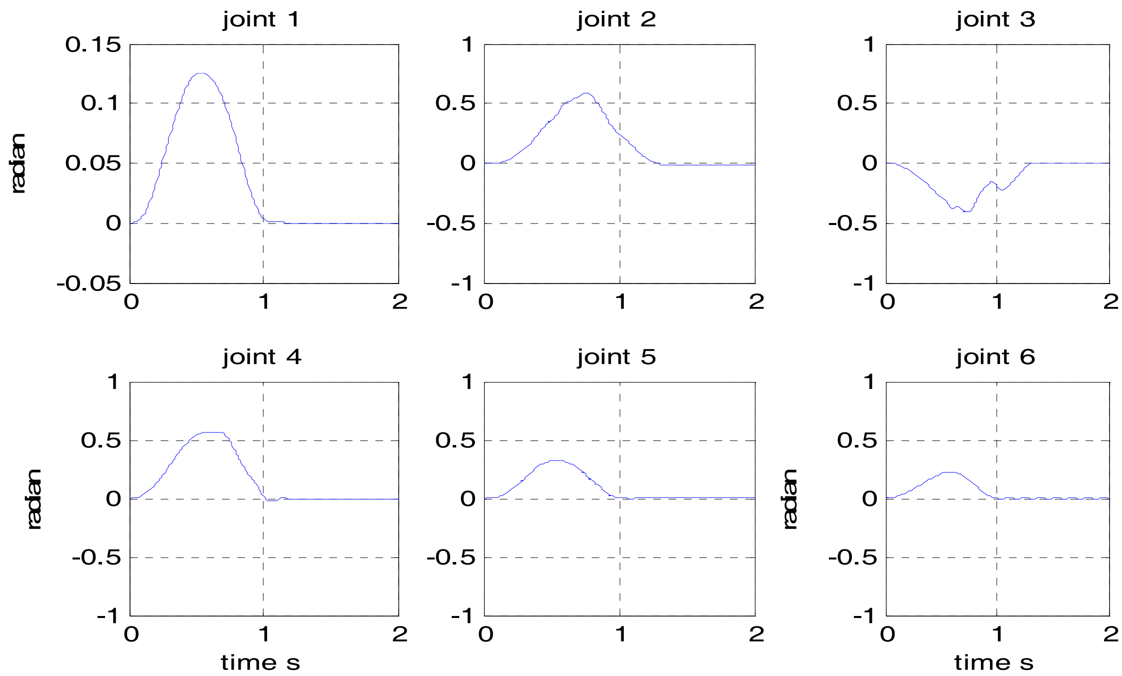

Figure 4c: Joint position tracing error torque under constrained linear MPC technique

HYBRID CONTROLLER:

In general, the main function of the first three joints in the robots is the positioning of the end-effector, whereas the last joints specifying its orientation. From Figures (3a, c and 4a, c) it is noticed that the IRp-6 achieves higher path tracking accuracy under CTC-PD controller than under MPC controller. However, MPC technique fulfilled the required process constraints. Furthermore, the wrist is the robot part which is in direct contact with the works (gripper, electrode). Therefore its movement constraints should be considered carefully by the controller. This consideration is realized only by applying MPC technique. Hence applying a hybrid system built from a linear CTC-PD controller for the first three joints and a linear MPC controller for the rest of joints will be a reasonable idea. The proposed scheme, its structure is shown in Figure (5), gathers the advantages of the two control strategies, the simplicity and low amount of calculation of CTC-PD technique and the capability of handling a multivariable constrained process of the MPC technique, in particular:

- Ensure accurate path tracking
- Guarantee positioning and orientation of the wrist of the robot with movement constraints compliance
- Higher computation efficiency

At each sampling time instant k equation (6) will be solved with the input torque vector, computed at each sampling instant $k, \tau(k) = [u_{pd}(k), u_{mpc}(k)]^T$, where $u_{pd}(k) \in \Re^3$ is the input vector from CTC-PD algorithm (7) and $u_{mpc}(k) \in \Re^3$ is the input vector computed from MPC algorithm (10-13). Therefore;

$$\begin{aligned} u_{pd}(k) &= M_0(1:3,1:3)(k_p \cdot e(k) + k_v \cdot \dot{e}(k)) + N_0(1:3,1) + G_0(1:3,1) + F_0(1:3,1) \\ &= [u_1(k); u_2(k); u_3(k)] \end{aligned} \tag{14}$$

$$u_{mpc}(k)=[u_4(k),u_5(k),u_6(k)]^T \quad (15)$$

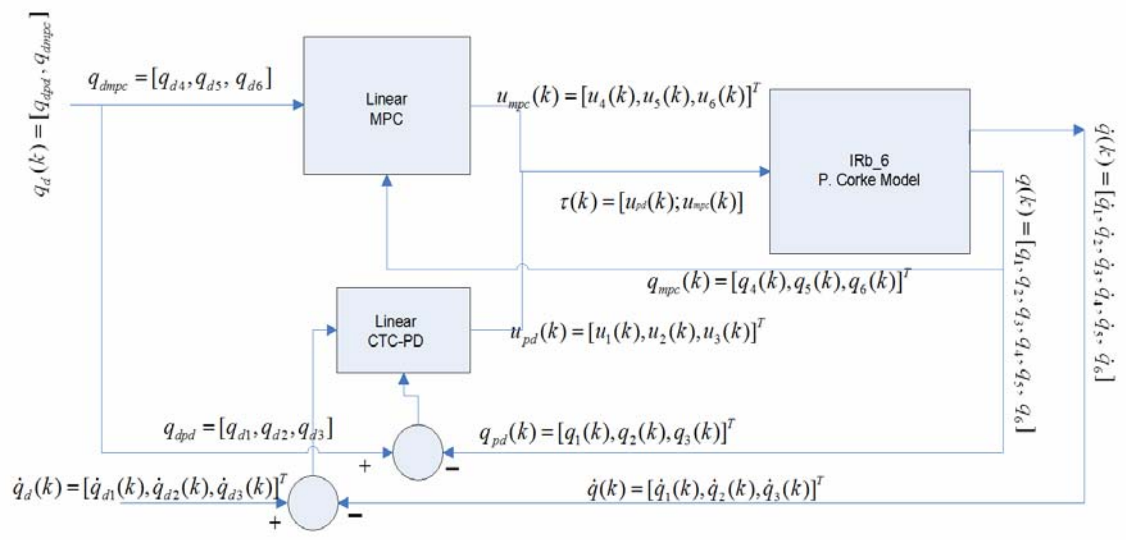

Figure 5: The linear hybrid controller structure

Result of simulation of IRp-6 operates under constrained hybrid technique:
Control parameters: N_p=40, N_u=6, sampling time T_s=0.01 s, λ =[100E-6; 200E-6; 500E-6], ψ =[800; 700; 1000], k_p =[1700; 1700; 20000] k_v =[48; 48; 56]
The input torque limits vector is (5), whereas output constraints for MPC algorithm are from Table (1).

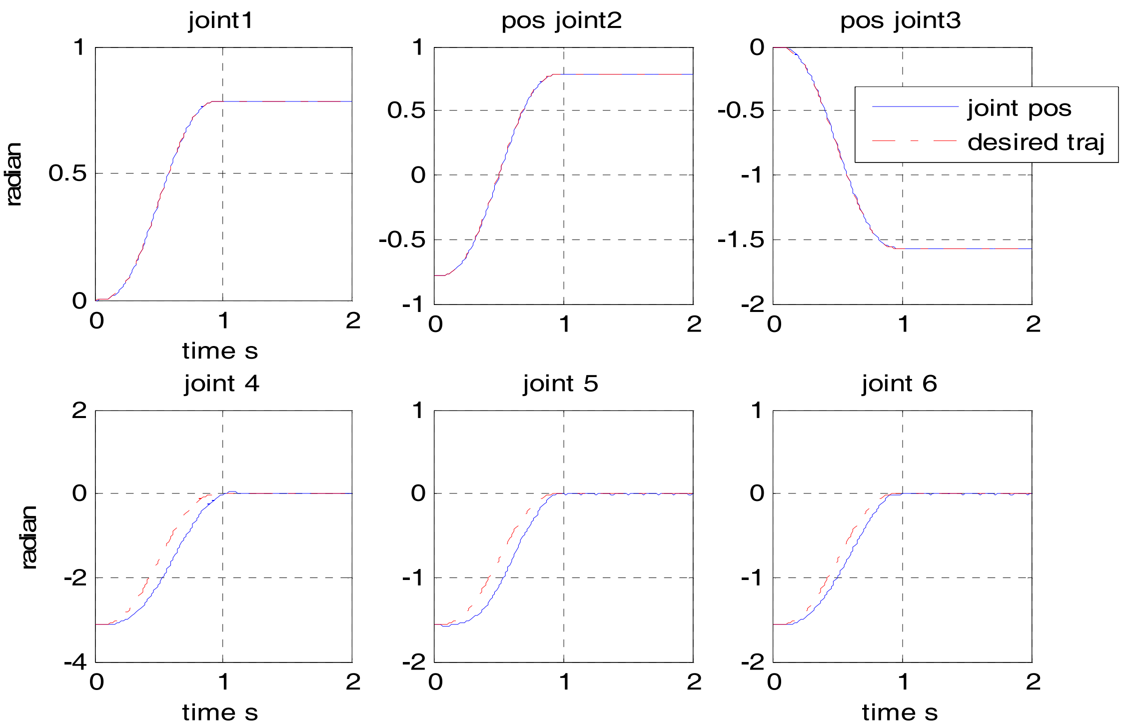

Figure 6a: Joints positions of IRp_6 robot under constrained linear hybrid controller

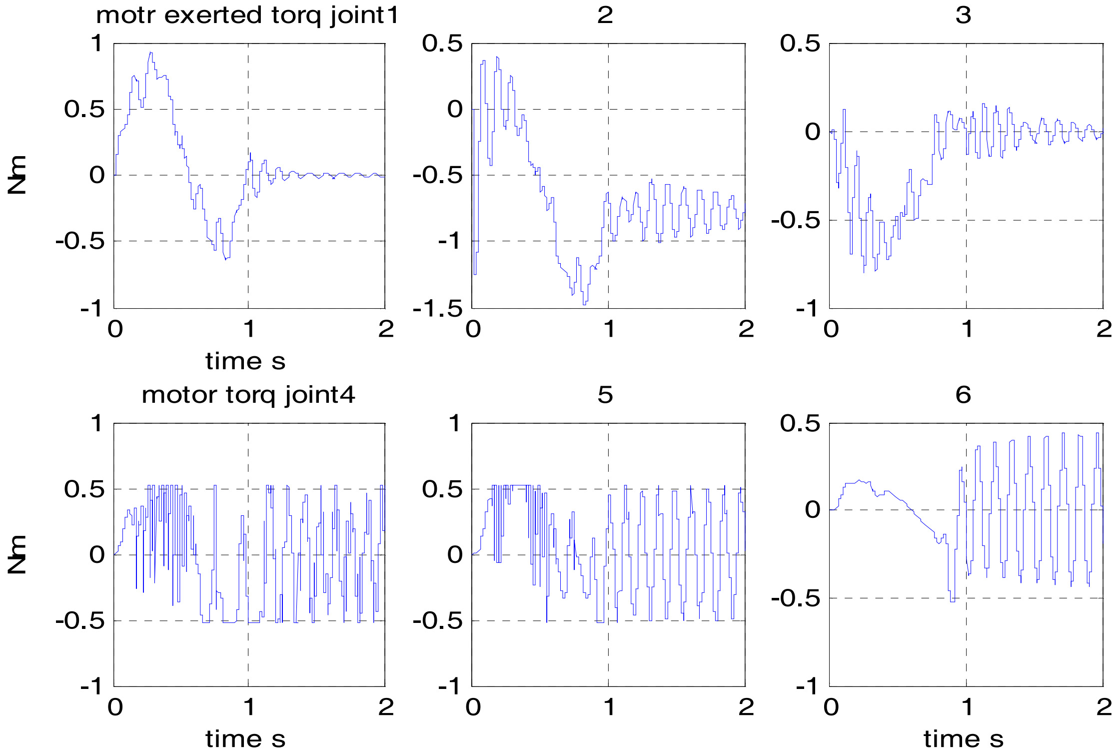

Figure 6b: IRb_6 joints input torque under constrained hybrid controller

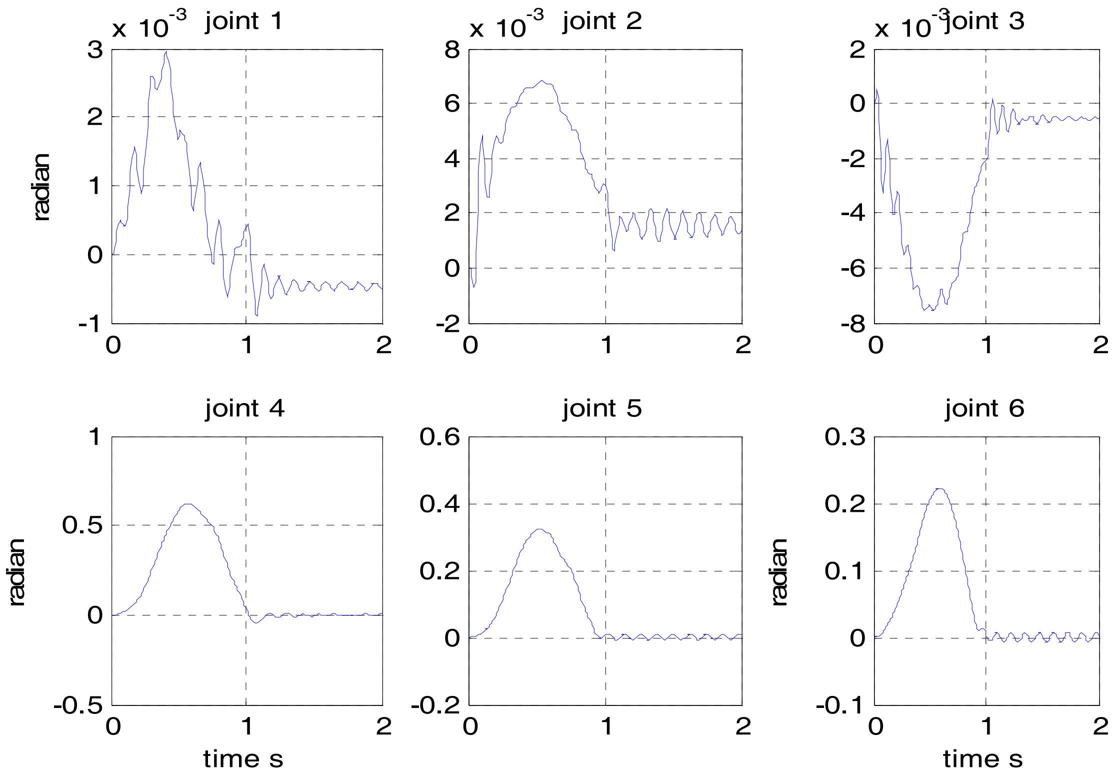

Figure 6c: IRb_6 robot joint position errors related to set-points under constrained hybrid controller

It is found that the execution time is 13, 40s.

CONCLUSION

In this paper, the IRp-6 robot object has been founded. Peter Corke model function toolboxes were used to generate the manipulator inertia matrix, Coriolis and centrifugal, friction and gravity vectors for joint space at one of the equilibrium configurations. Consequently, linear IRp-6 model was constructed. The joint space trajectories toolbox of 7^{th} order polynomial was used for generating the desired trajectories. The initial point joint space coordinates is the robot at rest (off workspace) and the final point is the robot at the goal point (on workspace). Both linear CTC-PD technique and linear MPC technique were applied to the IRp-6 robot. They achieve high position tracking. However, CTC-PD shows higher path tracing. Furthermore, a hybrid system composed from both control techniques was constructed. The linear CTC-PD technique controls the first three joints movements, whereas linear MPC technique applied for the rest of joints (the wrist). Simulation of the manipulator under this hybrid technique achieves high position tracking performance at the same time; it saves the system contacts by considering the wrist movement constraints. Furthermore, it saves over 26% of the computation effort compared with the application of MPC technique alone. The hybrid linear CTC-PD-MPC technique is a promising control strategy for multivariable fast systems such as industrial manipulators.

REFERENCES

[1] George Bekey and Junku Yuh, (2007) The Status of Robotics, report on thr WTEC international study, part II, digital object identifier 10-1109/M-RA 2007.907356.

[2] International Federation Robot (IFR), report 2009.

[3] M. Spong, S. Hutchinson and M. Vidyasagar, Wiley (2006), "Robot Modeling and control"

[4] S. Qin and T. Badgwell, "An Overview of Nonlinear Model Predictive Control Applications", Verlag, pp.369-392, 2000.

[5] Manfred Morari and Jay Lee, "Model predictive control: past, present and future", Computers and Chemical Engineering 23, 1999, pp 667-682

[6] S. Qin and T. Badgwell, "A survey of industrial model predictive control technology", control engineering practice 11, 2003, pp. 733-764

[7] Richalet j., Abu El Ata-Doss, Arber C., Kuntze H. B., Jacubasch A., Schill W., "Predictive Functional Control. Application to Fast and Accurate Robots" 10th IFAC World Congress, Munich 1997.

[8] Poignet Ph. And Gautier M., "Nonlinear Model Predictive Control of a Robot Manipulator", IEEE, 0-7803-5976-3, 2000, pp. 401-406.

[9] A. Vivas, V. Mosquera, "Predictive functional control of a PUMA robot", ACSE 05 Conference, 19-21 December 2005, CICC, Cairo Egypt.

[10] P. Tatjewski, Springs, London (2007), "Advanced Control of Industrial Processes structures and algorithms".

[11] J. M. Maciejowski, Prentice, Harlow (2002) "Predictive control with constraints"

[12] J. A. Rossiter, CRC Press (2003), "Model-Based Predictive Control, a Practical Approach".

[13] A. Benniran, "Superiority of MPC controller to CTC controller in Obstacle Avoidance with Application on IMI Direct Drive Revolute 2-DOF Manipulator", 4th International PhD Students and Young Scientists Conference, Warsaw Poland 21-23 September 2009.

[14] A. Benniran, “Application of nonlinear hybrid technique to IRp-6”, accepted in Automation 2010 Conference.
[15] P. I. Corke Robotic toolbox, www.cat.csiro.au/cmst/staff/pic/robot, April 2002.
[16] John Craig, Addison-Wesley (1986), “Robotics mechanics & control”.
[17] F. L. Lewis, D. W. Dawson and C. T. Abdullah Marcel, Dekker Inc.(2004), “Robot manipulator control theory and practice”.
[18] IRp_6 Manipulator Control Design study, by Vuong Dung under supervision of prof Zielinski Cezary (WUT) and prof Marcelo Aug (NUS) December 12th 2007
[19] A Gosiewski, K. Kurman, K. Malinowski, „checking the dynamics properties of Irb-6, subject PR-01.03.04 ” Instytut Automatyki Politechnika Warszawa, (December 1984)