World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:2, No:5, 2008

Dynamics and Control of Bouncing Ball

A. K. Kamath, N. M. Singh, and R. Pasumarthy

***Abstract*—**This paper investigates the control of a bouncing ball using Model Predictive Control. Bouncing ball is a benchmark problem for various rhythmic tasks such as juggling, walking, hopping and running. Humans develop intentions which may be perceived as our reference trajectory and tries to track it. The human brain optimizes the control effort needed to track its reference; this forms the central theme for control of bouncing ball in our investigations.

***Keywords*—**Bouncing Ball, impact dynamics, intermittent control, model predictive control.

I. Introduction

DYNAMICS and control of bouncing ball has been extensively studied and is a common model for numerous rhythmic tasks such as juggling ,walking, hopping and running. One degree of freedom juggler is nothing but a bouncing ball controlled via a sinusoidally actuated table. This can be perceived as a human controlling the bounce of a ball using a racquet / bat. Complex rhythmic tasks such as biped walking and locomotion of multiped robots are designed using juggling models. These models provide a theoretical basis for control of such rhythmic tasks.

The control technique applied in this paper is based on the belief that juggling, walking and hopping etc are human skills which are developed over a period of time. A juggler does not learn to juggle over a few attempts. He develops his skill over many attempts and during each he tries to master his inter-limb co-ordination and hand-eye co-ordination. The juggler first forms an intended trajectory of the juggled object in his mind, which can be perceived as a reference trajectory for the object. Using his skills, which have been mastered over many attempts, he tries to track his reference trajectory. An infant doesn't learn to walk overnight. It learns over a period of time developing its inter-limb and hand-eye co-ordination.

In short a human develops an intention to perform certain things in a certain manner before performing any task. He then uses his skills, which has developed over various practice efforts to get the desired result. During each attempt the human brain optimizes the control effort required to get the desired action. So the human through its actions acts as a Model Predictive Controller (MPC).

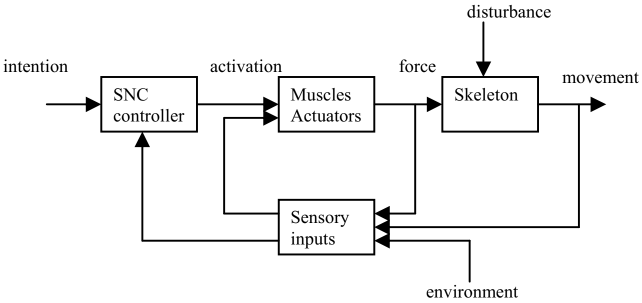


Fig. 1 Motor Sensory control in humans

This paper is organized as follows. Section II introduces the basic theory of MPC. We introduce the dynamics of bouncing ball in Section III. Section IV discusses about the implementation of control strategy. Stability and Robustness are discussed in Section V while results are presented in Section VI.

II. The Finite Horizon MPC Problem

Before getting to our benchmark example we discuss the basics of MPC in this section. Literature on MPC is abundantly available [4], [5], and [6]; here we briefly give an overview of it.

A. Model

Suppose a linear discrete-time, state space model of the plant is given in the form:

$$x(k+1) = Ax(k) + Bu(k) \tag{1}$$

$$y(k) = C_y x(k) \tag{2}$$

$$z(k) = C_z x(k) \tag{3}$$

where x is an n_x dimensional state vector, u is an n_u dimensional input vector, y is n_y dimensional vector of measured outputs and z is n_z dimensional vector of outputs which are to be controlled, either to a particular set-points, or to satisfy some constraints, or both. We assume y=z, and will then use C to denote both C_y and $C_{z.}$

A. K. Kamath is graduated student, Electrical Department, Veermata Jijabai Technological Institute, Matunga, Mumbai - 400019 India (e-mail: atulkamath@gmail.com).

N. M. Singh, Asst. Professor is with the Veermata Jijabai Technological Institute, Matunga, Mumbai -400019 India (e-mail: nmsingh59@gmail.com).

R. Pasumarthy is Research Fellow, Dept of Electrical and Electronic Engg, University of Melbourne, Australia (e-mail: r.pasumarthy@ee.unimelb.edi.au).

B. *Problem Formulation*

We shall assume that the plant model is linear, cost function is quadratic and there is no constraints i.e. unconstrained problem. Furthermore, we shall assume that the cost function does not penalize particular values of the input vector u(*k*), but only changes the input vector Δu(*k*), which is defined as:

$$\Delta u(k) = u(k) - u(k-1).$$

We shall not assume that the state variables can be measured, but that we can obtain an estimate $\hat{x}(k \mid k)$ of the state $x(k)$, the notation indicating that this estimate is based on measurements up to time k – that is, on measurements of the outputs up to y(k), and on knowledge of the inputs only up to u(k - 1), since the next input u(k) has not yet been determined. Signals $u(k+j \mid k)$ will denote a future value (at time k + j) of the input u, which is assumed at time k. Signals $x(k+j \mid k)$ and $y(k+j \mid k)$ will denote the predictions, made at time k, of the variables x and y at time k+j, on the assumption that some sequence of inputs $u(k+j \mid k)$ (i = 0; 1; : : : ; j ¡1) has been applied. These predictions will be made consistently with the assumed linearized model (1)–(3).

A cost function J penalizes deviations of the predicted controlled outputs $y(k+j \mid k)$ from a (vector-valued) reference trajectory $yref(k+j \mid k)$. Again the notation indicates that this reference trajectory may depend on measurements made up to time k; in particular, its initial point may be the output measurement y(k). But it may also be a fixed set-point, or some other predetermined trajectory. We define the cost function to be

$$Jk(x(k);u) = \sum_{j=1}^{N} \|y(k+j \mid k) - yref(k+j \mid k)\|^2_{Q(j)} + \sum_{j=0}^{Nc-1} \|\Delta u(k+j \mid k)\|^2_{R(j)} \qquad (4)$$

There are several points to note here. The prediction horizon has length N, but we do not necessarily have to start penalizing deviations of y from yref immediately, because there may be some delay between applying an input and seeing any effect. Nc is the control horizon. We will always assume here that Nc = N, i.e. the control horizon is equal to the prediction horizon. The form of the cost function (4) implies that the error vector y(k+j|k)-yref(k+j|k) is penalized at every point in the prediction horizon if Q(j) > 0. But it is possible to penalize the error at only a few coincidence points, by setting Q(j) = 0 for same values of j. It is also possible to have different coincidence points for different components of the error vector by setting appropriate elements of the weighting matrices Q(j) to 0. To allow for these possibilities, we do not insist on Q(j) > 0, but allow the weaker condition Q(j) ≥ 0. This condition is required, to ensure that Jk ≥ 0.

We also need R(j) ≥ 0 to ensure that Jk > 0. Again, we do not insist on the stronger condition that R(j) > 0, because there are cases in which the changes in the control signal are not penalized. The weighting matrices R(j) are sometimes called move suppression factors, since increasing them penalizes changes in the input vector more heavily.

C. *Prediction: Full State Measurement*

We deal with a simple situation here by assuming that the whole state vector is measured, so that $\hat{x}(k \mid k)$ = x(k) = y(k) (so C = I). Also assume that we know nothing about any disturbances or measurement noise. Then all we can do is to predict by iterating the model (1)–(3). So we get

$$\begin{aligned} x(k+j|k) &= A^j x(k) \\ &+ \begin{bmatrix} \sum_{i=0}^{j-1} A^i B & \ldots & B \end{bmatrix} \begin{bmatrix} \Delta u(k \mid k) \\ \cdot \\ \cdot \\ \Delta u(k+j-1) \end{bmatrix} \qquad (5) \\ &+ \sum_{i=0}^{j-1} A^i B u(k-1) \end{aligned}$$

for j ≤ N.

The prediction of y is now obtained as

$$y(k+j \mid k) = Cx(k+j \mid k) \qquad (6)$$

for j = 1;…. ;N.

D. *Solving the MPC Problem*

We can rewrite the objective function (4) as

$$Jk = \|Y(k) - Yref(k)\|^2{}_Q + \|\Delta U(k)\|^2{}_R \qquad (7)$$

where

$$Y(k) = \begin{bmatrix} y(K+1 \mid k) \\ : \\ y(k+N \mid k) \end{bmatrix} \quad Yref(k) = \begin{bmatrix} yref(K+1 \mid k) \\ : \\ yref(k+N \mid k) \end{bmatrix}$$

$$\Delta U(k) = \begin{bmatrix} \Delta u(k \mid k) \\ : \\ \Delta u(k+Nc-1 \mid k) \end{bmatrix}$$

And the weighting matrices Q and R are given by

$$Q = \begin{bmatrix} Q(1) & 0 & \cdots & 0 \\ 0 & Q(2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & & Q(N) \end{bmatrix}$$

$$R = \begin{bmatrix} R(1) & 0 & \cdots & 0 \\ 0 & R(2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & & R(Nc-1) \end{bmatrix}$$

From (5)-(6), we see that Y(k) has form

$$Y(k) = \Phi x(k) + \Gamma u(k\text{ -}1) + G_y \Delta U(k) \qquad (8)$$

For suitable matrices, Φ, Γ and G_y. Define

$$E(k) \;=\; Yref\text{ - }\Phi x(k) \;+\; \Gamma u(k\text{-}1) \qquad (9)$$

This vector can be thought of as a “tracking error”, in the sense that it is the difference between the future target trajectory and the “free response” of the system, namely the response that would occur over the prediction horizon if no input changes were made – that is, if we set ΔU(k) = 0. If E(k) really were 0, then it would indeed be correct to set ΔU(k) = 0. Now we can write

$$\begin{aligned} J_K &= \left\| G_y \Delta U(k) - E(k) \right\|^2_Q + \left\| \Delta U(k) \right\|^2_R \\ &= \left[\Delta U^T(k) G_y{}^T - E^T(k) \right] Q \left[G_y \Delta U(k) - E(k) \right] \\ &\quad + \Delta U^T(k) R \Delta U(k) \\ &= \Delta U^T(k) \left[G_y{}^T Q G_y + R \right] \Delta U(k) \\ &\quad - 2E^T(k) Q G_y \Delta U(k) + E^T(k) Q E(k) \end{aligned} \qquad (10)$$

This has the form

$$J_K = \frac{1}{2} \Delta U^T(k) H \Delta U(k) + f^T \Delta U(k) + const. \qquad (11)$$

where

$$H = 2(G_y{}^T Q G_y + R)$$

and

$$f = -2 G_y{}^T Q E(k)$$

and neither H nor f depends on ΔU(k).

The relationship between the input increments Δu and control input u is given by

$$\begin{bmatrix} u(k \mid k) \\ u(k+1 \mid k) \\ \vdots \\ \text{u(k} + Nc - 1) \end{bmatrix} = M \begin{bmatrix} \Delta u(k \mid k) \\ \Delta u(k+1 \mid k) \\ \vdots \\ \Delta \text{u(k + Nc - 1 | k)} \end{bmatrix} + Fu(k-1) \qquad (12)$$

where

$$M = \begin{bmatrix} I & 0 & \cdots & 0 \\ I & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I & I & \cdots & I \end{bmatrix}, \qquad \text{F} = \begin{bmatrix} \text{I} \\ \text{I} \\ \vdots \\ \text{I} \end{bmatrix}$$

So, from (11) we see that we have to solve the following unconstrained optimization problem

$$\min_{\Delta U(k)} \frac{1}{2} \Delta U^T(k) H \Delta U(k) + f^T \Delta U(k) \qquad (13)$$

This is a standard optimization problem known as the Quadratic Programming (QP) problem, and standard algorithms are available for its solution.

Remember that we use only the part of this solution corresponding to the first step, in accordance with the receding horizon strategy. So if the number of plant inputs is n_u then we just use the first n_u rows of the vector ΔUopt(k).

We can represent this as

$$\Delta u_{opt}(k) = \begin{bmatrix} I_{nu} & 0 & \ldots & 0 \end{bmatrix} \Delta U_{opt}(k)$$

III. Dynamics of Bouncing Ball

We use the model given by [3] which provides us with a two dimensional map for repeated impacts of a ball with a massive, sinusoidally vibrating table. We take the usual impact relationship.

$$V(t_j) - W(t_j) = -e(U(t_j) - W(t_j)) \qquad (14)$$

where U,V, and W are, respectively, the absolute velocities of the approaching ball, the departing ball and the table, 0<e≤1 is the co-efficient of restitution, and t = tj is the time of the jth impact. If we further assume that the distance the ball travels between impacts under the influence of gravity, g, is large compared with the overall displacement of the table, then the time interval between impacts is easily approximated as

$$t_{j+1} - t_j = \frac{2V(t_j)}{g}, \qquad (15)$$

and the velocity of approach at the (j+1)st impact as

$$U(t_{j+1}) = -V(t_j) \qquad (16)$$

Combining (14)-(16) and non-dimensionalizing, we obtain the recurrence relationship relating the state of the system at the (j+1)st impact to that at the jth in the form of a linear map.

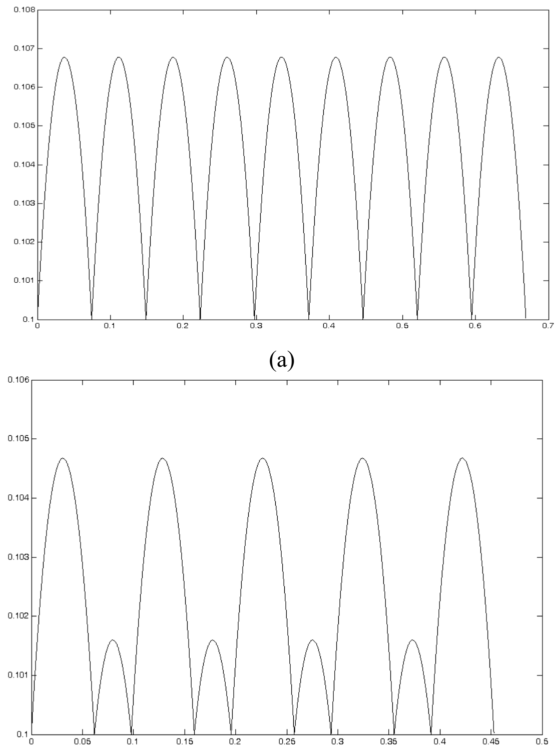
$$\begin{aligned} \upsilon_{j+1} &= e\upsilon_j + u_j \\ \phi_{j+1} &= \phi_j + \upsilon_j \end{aligned} \tag{17}$$

where $\phi = \omega t, \upsilon = 2\omega V / g$ and $u = 2\omega(1+e)W / g$

IV. Model Predictive Control of Bouncing Ball

From the dynamics of bouncing ball, we get a linear discrete-time, state space model of the form (1) where $x(k+1) = \upsilon_{j+1}, A = e, B = 1, u(k) = u_j$.

Since no control is applied to get ϕ_{j+1}, we consider here only velocity as we get a simple model which renders full state measurement.



(a)

(b)

Fig. 2 Reference trajectories (a) period one (b) period two

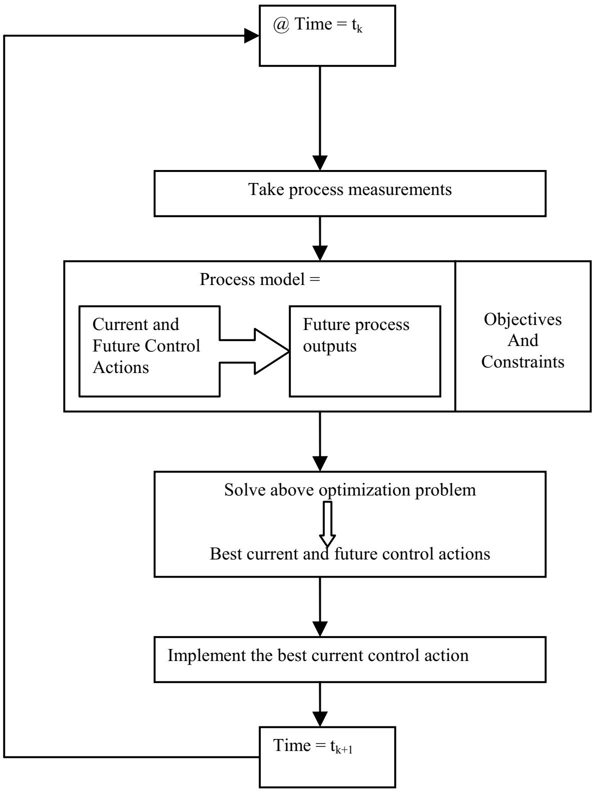

@ Time = t_k
Take process measurements
Process model =
Current and Future Control Actions
Future process outputs
Objectives And Constraints
Solve above optimization problem
Best current and future control actions
Implement the best current control action
Time = t_{k+1}

Fig. 3 Flowchart of Model Predictive Control

Following is the algorithm implemented to stabilize the bouncing ball around period one and two orbit.

1. Take state measurements using (8).
2. Calculate the tracking error using (9).
3. Solve the unconstrained optimization problem (13) to get a sequence of ΔU
4. Implement the current control action $\Delta U(k)$ using the control $u(k) = \Delta u(k) + u(k-1)$
5. Repeat steps 1-4.

V. Stability and Robustness

With a linear model and a quadratic objective, the resulting optimization problem takes the form of a highly structured convex Quadratic Program (QP) for which there exists a unique optimal solution. Several reliable standard solution codes are available for this problem.

Fig. 4 depicts the standard deviation of the normalized post-impact velocity v(k)/v* over 100 impacts, for increasing noise level. We observe that our controller obtains good noise rejection. This result is comparable with [1], where they compare the robustness with blind mirror law and piecewise quadratic laws with a suitable negative acceleration.

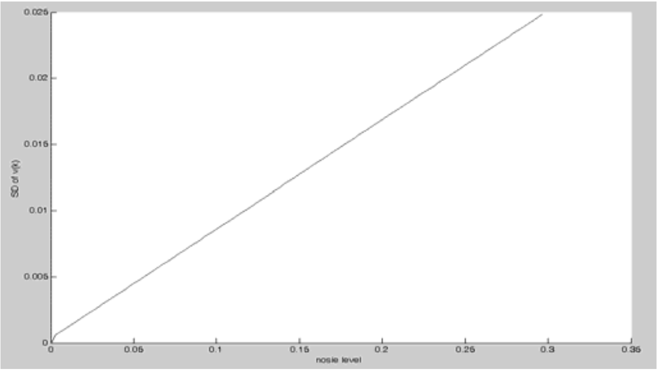


Fig. 4 Simulation result of a noisy bouncing ball dynamics

VI. Simulation Results

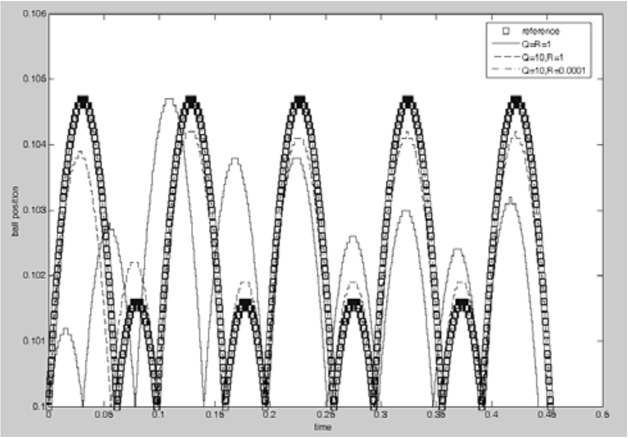


Fig. 5 Period 2 stabilization of bouncing ball. Reference trajectory is depicted by square. For Q=R=1trajectory is depicted by solid line. For Q=10, R=1 is depicted by dashed line while for Q=10, R=0.001 is depicted by dash dot line

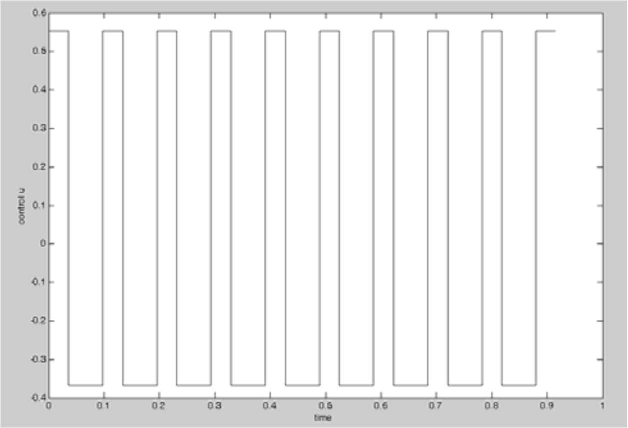


Fig. 6 Control input applied for achieving period 2. Control input is constant over a two successive impacts

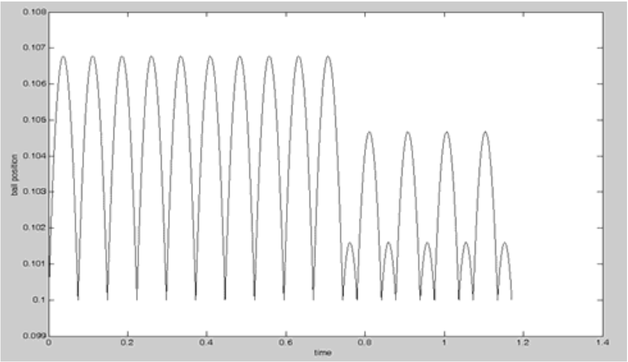


Fig. 7 Bouncing ball from period one to period two

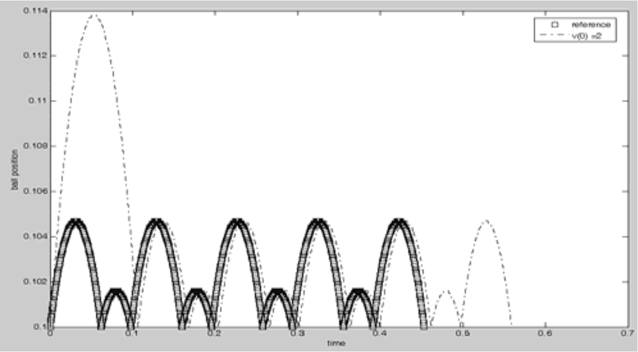


Fig. 8 Ball tracking period two trajectory with initial velocity v(1) = 2

VII. Conclusion and Future Work

We have started applying the control paradigm presented in this paper to other rhythmic tasks like juggler and biped. Encouraging results have been obtained which will be reported in the future.

References

[1] R. Ronsse, P. Lefevre, and R. Sepulchre,, "Rhythmic Feedback Control of a Blind Planar Juggler," IEEE Transactions on Robotics, Vol.24,No.4., pp. 790-802, August 2007.

[2] R. Ronsse, P. Lefevre, and R. Sepulchre, "Sensorless stabilization of bounce juggling," *IEEE Trans. Robot.*, vol. 22, no. 1, pp. 147–159, Feb. 2006.

[3] J. Guckenheimer and P. J. Holmes, *Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields*. New York: Springer-Verlag, 1986.

[4] A. Tiagounov J. Buijs, S. Weiland and B. De Moor. Long horizon model predictive control for nonlinear industrial processes. In Proc. European Control Conference, Cambridge, UK, 2003.

[5] A. Tiagounov, D. van Hessem, M. Balenovic and S. Weiland. From state estimation to long horizon MPC for nonlinear industrial , applications. In INCOOP Workshop materials, D¨usseldorf, Germany, 2003.

[6] A. Tiagounov and S. Weiland. Model predictive control algorithm for nonlinear chemical processes. In Proc. Physics and Control Conference, St.Petersburg, Russia, 2003.

[7] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality", in Automatica 36 (2000) pp.789-814.