Third International Conference on
Advances in Control and Optimization of Dynamical Systems
March 13-15, 2014. Kanpur, India

Computationally Efficient Suboptimal Control design for Impulsive Systems based on Model Predictive Static Programming

Chandrashekhar M. Sakode * **Radhakant Padhi** **

* *(Research Scholar, Indian Institute of Science, Bangalore*
** *Associate Professor, Indian Institute of Science, Bangalore (e-mail: padhi@aero.iisc.ernet.in)*

Abstract: A new suboptimal control design approach for impulsive control system is proposed. It is extension of model predictive static programming for continuous control system. This approach is applicable to finite time problems with terminal constraints. Starting from initial guess control history, control is updated in iterative manner till convergence criteria is met. It is computationally efficient, hence, can be implemented online and it gives closed form control solution when control is unconstrained. Also, it does not require approximation of system dynamics. As an example problem, predator-prey (Lotka Volterra), which is a nonlinear model is considered, and simulation results are shown. System states are driven to its equilibrium point. Here fish and shark harvesting is represented in the form of impulse control. Harvesting is done in 10 equal intervals (9 times a year), whereas plots are shown for two years. It took 1-2 sec to run this algorithm.

1. INTRODUCTION

If there is instantaneous change of least one state variable with application of control input, then such systems are called as impulsive or impulse control systems (Yang [2001]). Also, control input/s are applied at some points of time (impulse instants) and not continuously. Impulse instants can be fixed a priori at equal or unequal interval or it can be part of control design variable alongwith control magnitude. Sometimes impulse control is preferred over continuous control because continuous control is not applicable or impulse control is more efficient or cheaper or practical. There are various approaches to solve impulse control problem. For example, in Kemih and Chahim [2012] impulse control maximum principle is used and resulting multipoint boundary value problem is solved. In Yang et al. [2010], particle swarm optimization is used. In Hou and Wong [2011], Pontryagin maximum principle is used and optimization problem is formulated at impulse instants, which is solved using Matlab Fmincon. In Wong and Balakrishnan [2010], approximate dynamic programming (ADP) is used and resulting optimal control equations are solved by using Single network adaptive critic (SNAC). All of these methods are computationally inefficient.

There are two approaches for solving optimal control problems, indirect and direct methods. In direct method, system is discretized and optimization problem is formulated with state and control as optimization variables with different constraints (Betts [2001]). This approach is computationally intensive. In indirect method there are two main philosophies, i.e., Pontryagin maximum principle and dynamic programming (Kirk [2004]), here first necessary conditions for optimal control are derived, which are solved to get optimal control. Pontryagin principle leads to two point boundary value problem (TPBVP) in which state and costate dynamics have to be solved in forward and backward direction respectively, as initial value of state and final value of costate are available. This again is computationally intensive. Also, costate variable is not physical variable hence, it is difficult to assume. Dynamic programming leads to nonlinear partial differential equation (PDE)(which are difficult to solve) or leads to curse of dimensionality (because of discretization). To avoid curse of dimensionality in dynamic programming, approximate dynamic programming (ADP) is used in literature (Werbos [1992]). ADP is again discrete formulation, it is combined with heuristic programming and necessary conditions of optimal controls are derived. Neural network strategies are used to solve these necessary conditions which gives control solution in state feedback form. This approach is computationally intensive and it also requires assumption of costate as a function of state. Another approach called as model predictive control (MPC) is widely used in industry (like chemical processes which are slow). Here, optimization problem is formulated where cost function is based on output and control, other constraints are taken in discretized form. This has a computational advantage over cost function with states, because output variables are lesser in dimension than state variables. However, other state variables which are not part of output variable/s must remain stable.

In this work, we propose a computationally efficient new algorithm for impulsive control system based on model predictive static programming (MPSP). This work is extension of Padhi and Kothari [2009], where MPSP was proposed for continuous control system. MPSP is based on the philosophy of ADP and MPC. In MPSP similar to MPC, optimization problem is formulated and control is updated iteratively to minimize ouput error. Similar to

ADP, where discretized dynamics is taken and costate is propagated in backward direction, in MPSP discretized dynamics is taken and output error is propagated in backward direction. MPSP for impulse control is a suboptimal control design with terminal constraints. Here, output error at final time is expressed in terms of control deviation using discretized state dynamics. Optimization problem is defined to minimize quadratic cost function of control subject to constraints of output error which is expressed in control deviation. MPSP with impulse control has several advantages: i) terminal constraints are met; ii) TPBVP are avoided. iii) computationally efficient algorithm because some of the matrices can be computed recursively, which are used for computation of control and control has to be computed at the time of impulse only and not at all time steps; iv) nonlinear dynamics is used without any approximation; v) there is no need to assume value for any nonphysical variable like costate variable as in Pontryagin maximum principle or in ADP. Here, only Lagrange multiplier is required, which can be found without any assumption by standard optimization techniques. For verification of MPSP with impulse control, Lotka Volterra (predator-prey) model with impulse control is considered as an example problem. Lotka Volterra is standard vector nonlinear problem.

2. MODEL PREDICTIVE STATIC PROGRAMMING FOR IMPULSIVE CONTROL SYSTEM

Consider a nonlinear impulsive control system,

$$\dot{X} = f(X), \quad \text{for } t \neq t_q \text{ and } q = 1, 2, \ldots, n_q \tag{1}$$

$$X_k^+ = g\left(X_k^-, U_q\right), \quad \text{for } t = t_q \tag{2}$$

$$Y = h(X) \tag{3}$$

$$X(0) = X_0 \tag{4}$$

here, $X \in \Re^n$ are states, $f: \Re^n \rightarrow \Re^n$ is a nonlinear function, t_q is time at q^{th} impulse, n_q is the total number of impulses, $U_q \in \Re^m$ are impulse controls at time t_q, $g: \Re^n \times \Re^m \rightarrow \Re^n$ is nonlinear function, X_k^- and X_k^+ are state values just before and after impulse at t_q i.e., U_q respectively, $Y \in \Re^p$ are outputs which are to be controlled, $h: \Re^n \rightarrow \Re^p$ is nonlinear function, and X_0 is initial value of states at time $t = 0$. Let t_0 be initial time and t_f be final time. Let Y^d be the desired value of Y, which is to be achieved at final time t_f i.e., $Y \rightarrow Y^d$ at t_f. MPSP needs discrete formulation, therefore discretizing t_0 to t_f uniformly in N discrete time instants including initial and final time. At impulse instants $t_1, t_2, \ldots, t_{n_q}$, there is instantaneous change in state values because of impulse controls $U_1, U_2, \ldots, U_{n_q}$ respectively. Therefore, at impulse instants state will have two values at same point of time, hence, each state trajectory can be divided in number of segments. Total number of segments, n_{seg} are given by, $n_{seg} = n_q + 1$, also it is assumed that if more than one controllers are present then all controllers will operate simultaneously, control is not applied at initial and final time. Let n_d be number of nodes per segment for each state. Last node and first node of adjacent segments will have same impulse instant. For example, last node of first segment and first node of second segment will have same impulse instant time t_1 and last node of second segment and first node of third segment will have same impulse instant time t_2 and so on. Note that $n_d n_{seg}$ are total number of distinct state values for each state and N is the total number of distinct time instants. $N < n_d n_{seg}$ because, at impulse instants, state will have two values. Also, $n_q \ll N$ as less number of impulses will be applied compared to distinct time instants. Relationship between n_q, N, n_d, and n_{seg} is given by,

$$N + n_q = n_d n_{seg} \tag{5}$$

Consider an example shown in Fig.1, whole time interval is divided in 10 discrete time instants thus $N = 10$. There is one state X and one control u with two impulses, u_1 and u_2. Thus, total number of impulses $n_q = 2$ and total number of segments, $n_{seg} = n_q + 1 = 3$. Total number of nodes per segment n_d is, $n_d = 4$. Each segment node numbers are given as 1,2,3,4. Initial state X_0 is first node of first segment. Time instants 4 and 7 are impulse instants, at each of these impulse instants state has two values, X_4^-, X_4^+ and X_7^-, X_7^+. Also, $N + n_q = 12$ and $n_d n_{seg} = 12$ thus, $N + n_q = n_d n_{seg}$.

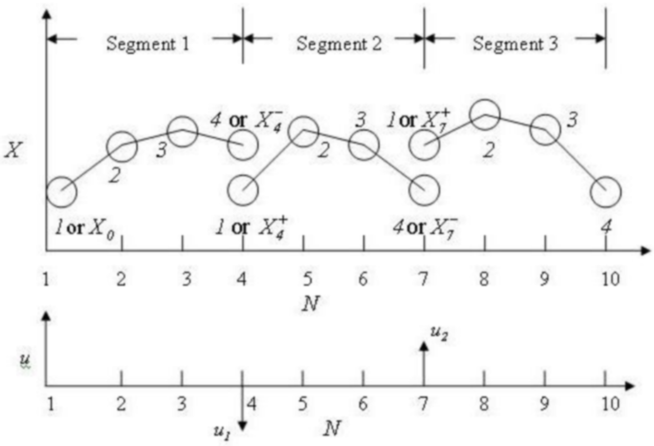

Fig. 1. Example showing application of impulse control on discretized states with different segments and nodes.

Equation (1) is discretized using Euler equation i.e., $X_{k+1} = X_k + \Delta t f(X_k)$, Δt is step size, (2) is already in discretized form and (3) is discretized as $Y_N = h(X_N)$, as we are interested in output at final time. Note that $X_N = X(t_f)$ and $Y_N = Y(t_f)$. Discretized form can be written as,

$$X_{k+1} = F(X_k), \quad \text{for } k = 1, 2, \ldots, N \tag{6}$$

$$X_k^+ = g\left(X_k^-, U_q\right), \quad \text{for } t = t_q, \quad q = 1, 2, \ldots, n_q \tag{7}$$

here, q represents impulse instants and k represents different instants including impulse instants. Therefore, at impulse instants k and q will be related. For example, q^{th} impulse will be applied to last node of q^{th} segment. Therefore, impulse instants, k and q are related as,

$$k + q = n_d q + 1 \tag{8}$$

note that (8) is slightly different from (5). Equation (5) is relationship between total number of impulses, n_q, total number of segments, n_{seg} and total time instants, N, whereas (8) is relationship between, q^{th} impulse which is applied on k^{th} node. Rearranging (8) we get,

$$q = \left(\frac{k-1}{n_d - 1}\right) \tag{9}$$

$$Y_N = h(X_N) \tag{10}$$

In discretized form control objective is $Y_N \rightarrow Y^d$. Initial control history is assumed at n_q impulse instants and state is propagated using (6) and (7). Hence, by using (10) Y_N can be obtained from X_N. Let $\Delta Y_N = Y^d - Y_N$ be the error in output at final time. Also assuming that initial control history is such that Y_N is close to Y^d, therefore, $\Delta Y_N \approx dY_N$. Therefore, to begin with good initial guess control history is required, however initial control history will not satisfy objective $Y_N \rightarrow Y^d$. dY_N is used to compute the control at impulse instants in iterative manner, i.e., if error criteria is not met (dY_N is not close to zero) then control history is updated iteratively till the convergence. Let X_N^d be desired value of X which gives Y^d i.e., $Y^d = h\left(X_N^d\right)$. It is assumed that if $X_N \rightarrow X_N^d$, then $Y_N \rightarrow Y^d$ i.e., if $X_N \approx X_N^d$ then $Y_N \approx Y^d$. Taylor series expansion of (10) about X_N with first order terms gives,

$$Y^d = Y_N + \left[\frac{\partial Y_N}{\partial X_N}\right]_{X_N} \left(X_N^d - X_N\right) \tag{11}$$

therefore,

$$dY_N = \left[\frac{\partial Y_N}{\partial X_N}\right]_{X_N} dX_N \tag{12}$$

here, $dY_N = Y^d - Y_N$ and $dX_N = X_N^d - X_N$ (assuming X_N is close to X_N^d). Similarly from (6), Taylor series expansion about with first order terms gives,

$$X_{k+1}^d = X_{k+1} + \left[\frac{\partial F(X_k)}{\partial X_k}\right]_{X_k} \left(X_k^d - X_k\right) \tag{13}$$

here, X_{k+1}^d is desired state value at $(k+1)^{th}$ time instant and X_{k+1} is state value at $(k+1)^{th}$ time instant. X_k^d is desired state value at $(k)^{th}$ time instant and X_k is state value at $(k)^{th}$ time instant. From (13),

$$dX_{k+1} = \left[\frac{\partial F(X_k)}{\partial X_k}\right]_{X_k} dX_k \tag{14}$$

here, $dX_{k+1} = X_{k+1}^d - X_{k+1}$ (assuming X_{k+1} is close to X_{k+1}^d) and $dX_k = X_k^d - X_k$ (assuming X_k is close to X_k^d). Thus, dX_{k+1} is function of dX_k, similarly, dX_N will be function of dX_{N-1}, dX_{N-1} will be function of dX_{N-2} and so on. Equation (14) is substituted in (12) with $k = N-1$, thus we get,

$$dY_N = \left[\frac{\partial Y_N}{\partial X_N}\right] \left[\frac{\partial F(X_{N-1})}{\partial X_{N-1}}\right] dX_{N-1} \tag{15}$$

again expanding dX_{N-1} as a function of dX_{N-2} by using (14), we get,

$$dY_N = \left[\frac{\partial Y_N}{\partial X_N}\right] \left[\frac{\partial F(X_{N-1})}{\partial X_{N-1}}\right] \left[\frac{\partial F(X_{N-2})}{\partial X_{N-2}}\right] dX_{N-2} \tag{16}$$

similarly, expanding in this manner upto n_k^{th} impulse, (note that we have started by taking output error at final time, dY_N, and we are moving in backward direction therefore, n_k^{th} impulse will come first in backward direction), therefore we get,

$$\begin{aligned} dY_N = & \left[\frac{\partial Y_N}{\partial X_N}\right] \left[\frac{\partial F(X_{N-1})}{\partial X_{N-1}}\right] \cdots \\ & \cdots \left[\frac{\partial F\left(X_{N-(n_d-1)}^+\right)}{\partial X_{N-(n_d-1)}^+}\right] dX_{N-(n_d-1)}^+ \end{aligned} \tag{17}$$

taking first order Taylor series expansion of (7) about X_k^- and U_q we get,

$$\begin{aligned} X_k^{+d} = & X_k^+ + \left[\frac{\partial g\left(X_k^-, U_q\right)}{\partial X_k^-}\right]_{X_k^-} \left(X_k^{-d} - X_k^-\right) + \\ & \left[\frac{\partial g\left(X_k^-, U_q\right)}{\partial U_q}\right]_{U_q} \left(U_q^d - U_q\right) \end{aligned} \tag{18}$$

Here, U_q^d is desired value of control at q^{th} impulse instant and U_q is control value at q^{th} impulse instant. Let, $dX_k^+ = X_k^{+d} - X_k^+$, $dX_k^- = X_k^{-d} - X_k^-$ and $dU_q = U_q^d - U_q$ (assuming U_q close to U_q^d), using it in (18) gives,

$$\begin{aligned} dX_k^+ = & \left[\frac{\partial g\left(X_k^-, U_q\right)}{\partial X_k^-}\right]_{X_k^-} dX_k^- + \\ & \left[\frac{\partial g\left(X_k^-, U_q\right)}{\partial U_q}\right]_{U_q} dU_q \end{aligned} \tag{19}$$

for $q = 1, 2, \ldots, n_q$. Using equation (19) in equation (17) with $k = N - (n_d - 1)$ and correspondingly, $q = n_q$ (note that corresponding value of q at a particular impulse instant k can also be found out by using (9)) gives,

$$\begin{aligned} dY_N = & \left[\frac{\partial Y_N}{\partial X_N}\right] \left[\frac{\partial F}{\partial X_{N-1}}\right] \cdots \left[\frac{\partial F}{\partial X_{N-(n_d-1)}^+}\right] \times \\ & \left[\left[\frac{\partial g}{\partial X_{N-(n_d-1)}^-}\right] dX_{N-(n_d-1)}^- + \left[\frac{\partial g}{\partial U_{n_q}}\right] dU_{n_q}\right] \end{aligned} \tag{20}$$

here, some arguments are dropped for brevity. Rearranging (20),

$$\begin{aligned} dY_N = & \left[\frac{\partial Y_N}{\partial X_N}\right] \left[\frac{\partial F}{\partial X_{N-1}}\right] \cdots \left[\frac{\partial F}{\partial X_{N-(n_d-1)}^+}\right] \\ & \times \left[\frac{\partial g}{\partial X_{N-(n_d-1)}^-}\right] dX_{N-(n_d-1)}^- + \left[\frac{\partial Y_N}{\partial X_N}\right] \times \\ & \left[\frac{\partial F}{\partial X_{N-1}}\right] \cdots \left[\frac{\partial F}{\partial X_{N-(n_d-1)}^+}\right] \left[\frac{\partial g}{\partial U_{n_q}}\right] dU_{n_q} \end{aligned} \tag{21}$$

using (14) to expand $dX_{N-(n_d-1)}^-$ as function of $dX_{N-(n_d-1)-1}$ therefore, (21) becomes,

$$\begin{aligned} dY_N = & \left[\frac{\partial Y_N}{\partial X_N}\right] \left[\frac{\partial F}{\partial X_{N-1}}\right] \cdots \left[\frac{\partial F}{\partial X_{N-(n_d-1)}^+}\right] \times \\ & \left[\frac{\partial g}{\partial X_{N-(n_d-1)}^-}\right] \left[\frac{\partial F}{\partial X_{N-(n_d-1)-1}}\right] dX_{N-(n_d-1)-1} \\ & + \left[\frac{\partial Y_N}{\partial X_N}\right] \left[\frac{\partial F}{\partial X_{N-1}}\right] \cdots \left[\frac{\partial F}{\partial X_{N-(n_d-1)}^+}\right] \times \\ & \left[\frac{\partial g}{\partial U_{n_q}}\right] dU_{n_q} \end{aligned} \tag{22}$$

again by using (14) $dX_{N-(n_d-1)-1}$ can be expanded as a function of $dX_{N-(n_d-1)-2}$ and so on upto $(n_q-1)^{th}$ impulse, therefore, (22) becomes,

$$\begin{aligned} dY_N = & \left[\frac{\partial Y_N}{\partial X_N}\right]\left[\frac{\partial F}{\partial X_{N-1}}\right]\cdots\left[\frac{\partial F}{\partial X^+_{N-(n_d-1)}}\right]\times \\ & \left[\frac{\partial g}{\partial X^-_{N-(n_d-1)}}\right]\left[\frac{\partial F}{\partial X_{N-(n_d-1)-1}}\right]\cdots \\ & \left[\frac{\partial F}{\partial X^+_{N-2(n_d-1)}}\right] dX^+_{N-2(n_d-1)} + \left[\frac{\partial Y_N}{\partial X_N}\right] \\ & \times\left[\frac{\partial F}{\partial X_{N-1}}\right]\cdots\left[\frac{\partial F}{\partial X^+_{N-(n_d-1)}}\right]\left[\frac{\partial g}{\partial U_{n_q}}\right] dU_{n_q} \end{aligned} \tag{23}$$

$k = N - 2(n_d - 1)$ is impulse instant and here (19) will be used to replace, $dX^+_{N-2(n_d-1)}$ and corresponding q is $n_q - 1$, therefore, (23) becomes,

$$\begin{aligned} dY_N = & \left[\frac{\partial Y_N}{\partial X_N}\right]\left[\frac{\partial F}{\partial X_{N-1}}\right]\cdots\left[\frac{\partial F}{\partial X^+_{N-(n_d-1)}}\right]\times \\ & \left[\frac{\partial g}{\partial X^-_{N-(n_d-1)}}\right]\cdots\left[\frac{\partial F}{\partial X^+_{N-2(n_d-1)}}\right] \\ & \times\left[\frac{\partial g}{\partial X^-_{N-2(n_d-1)}}\right] dX^-_{N-2(n_d-1)} + \left[\frac{\partial Y_N}{\partial X_N}\right]\times \\ & \left[\frac{\partial F}{\partial X_{N-1}}\right]\cdots\left[\frac{\partial F}{\partial X^+_{N-(n_d-1)}}\right]\left[\frac{\partial g}{\partial X^-_{N-(n_d-1)}}\right] \\ & \times\left[\frac{\partial F}{\partial X_{N-(n_d-1)-1}}\right]\cdots\left[\frac{\partial F}{\partial X^+_{N-2(n_d-1)}}\right]\times \\ & \left[\frac{\partial g}{\partial U_{n_{q-1}}}\right] dU_{n_{q-1}} + \left[\frac{\partial Y_N}{\partial X_N}\right]\left[\frac{\partial F}{\partial X_{N-1}}\right]\cdots \\ & \cdots\left[\frac{\partial F}{\partial X^+_{N-(n_d-1)}}\right]\left[\frac{\partial g}{\partial U_{n_q}}\right] dU_{n_q} \end{aligned} \tag{24}$$

similarly, dX^-_k are expanded using (14) upto adjacent impulse instant, then dX^+_k is expanded using (19) and so on upto 1^{st} impulse. At 1^{st} impulse dX^-_k will be $dX^-_{N-n_q(n_d-1)}$ ($k = N - n_q(n_d-1)$ is location of 1^{st} impulse instant). Expansion of $dX^-_{N-n_q(n_d-1)}$ using (14) will lead to,

$$dX^-_{N-n_q(n_d-1)} = \left[\frac{\partial F}{\partial X_{N-n_q(n_d-1)}}\right]\cdots\left[\frac{\partial F}{\partial X_1}\right] dX_1 \tag{25}$$

here, $dX_1 = dX_0 = 0$, as first node of first segment is initial condition and there is no change in initial condition. Thus, finally (24) will contain only dU terms and it is written as,

$$B_1 dU_1 + B_2 dU_2 + \ldots + B_q dU_q = dY_N \tag{26}$$

q is the number of impulses. where,

$$\begin{aligned} B_q = & \left[\frac{\partial Y_N}{\partial X_N}\right]\left[\frac{\partial F}{\partial X_{N-1}}\right]\cdots\left[\frac{\partial F}{\partial X^+_{N-(n_d-1)}}\right] \\ & \times\left[\frac{\partial g}{\partial X^-_{N-(n_d-1)}}\right]\left[\frac{\partial F}{\partial X_{N-(n_d-1)-1}}\right]\cdots \\ & \cdots\left[\frac{\partial F}{\partial X^+_{N-[n_q-(q-1)](n_d-1)}}\right]\left[\frac{\partial g}{\partial U_q}\right] \end{aligned} \tag{27}$$

For computation of B_q, multiplication of $[n_q - (q-1)]\, n_d + 1$ matrices are required which will be large number. For example, if $n_q = 5$ and $n_d = 50$, then for computation of B_1, product of 251 matrices will be required. However, $B'_q s$ are can be computed recursively because some of the matrices are same in different $B'_q s$. For example, in (24), matrices multiplied to dU_{n_q} except $\left[\frac{\partial g}{\partial U_{n_q}}\right]$ are also multiplied with dU_{n_q-1}. Recursive relation for B_q is given below,

$$B^0_{n_q} = \left[\frac{\partial Y_N}{\partial X_N}\right]\left[\frac{\partial F}{\partial X_{N-1}}\right]\cdots\left[\frac{\partial F}{\partial X^+_{N-(n_d-1)}}\right] \tag{28}$$

$$B_q = B^0_q\left[\frac{\partial g}{\partial U_q}\right] \tag{29}$$

for $q = 1, 2, \ldots, n_q$

$$\begin{aligned} B^0_q = & B^0_{q+1}\left[\frac{\partial g}{\partial X^-_{N-[n_q-((q+1)-1)](n_d-1)}}\right]\times \\ & \left[\frac{\partial F}{\partial X_{N-[n_q-((q+1)-1)](n_d-1)-1}}\right]\cdots \\ & \cdots\left[\frac{\partial F}{\partial X^+_{N-[n_q-(q-1)](n_d-1)}}\right] \end{aligned} \tag{30}$$

for $q = 1, 2, \ldots, n_q - 1$ Recursive matrices computation is described as, first compute $B^0_{n_q}$ by using (28) and use it in (29) to compute B_{n_q}. Now $B^0_{n_q-1}$ is computed from $B^0_{n_q}$ by using (30) , then by using (29) $B_{n_{q-1}}$ is computed. Now, $B^0_{n_{q-2}}$ is computed from $B^0_{n_{q-1}}$ by using (30) and so on, finally B_1 is computed. $B'_q s$ are called as sensitivity matrices. From (26), if number of ouput variables are less than number of independent controls i.e., $p < mn_q$ then additional control objectives can be achieved and hence optimization problem can be formulated as described below,

$$Minimize\ J = \left(\frac{1}{2}\right)\sum_{q=1}^{n_q}(U_q + dU_q)^T R_q (U_q + dU_q) \tag{31}$$

subject to

$$\sum_{q=1}^{n_q} B_q dU_q = dY_N \tag{32}$$

where B_q is given by (27) ((32) is same as (26)) and (31) is control minimization problem with dU_q as optimization variable, U_q for $q = 1, 2, \ldots, n_q$ as control history and R_q is positive definite matrix for weighting different control elements. Note that $U_q + dU_q$ is control value because as mentioned earlier, $dU_q = U^d - U_q$, therefore, $U^d = U_q +$

dU_q. Hence, if we start from some control history U_q and to reach U^d, update of dU_q will be required. Thus, $U_q + dU_q$ is control value which is to be minimized. MPSP is iterative algorithm and U_q will be updated by dU_q till convergence of dY_N. Equations (31) and (32) are standard static optimization problem which can be solved by Lagrange multiplier method (Rao [2009]). Solution of (31) and (32) is described. Forming an adjoint equation,

$$\begin{aligned}\bar{J} = \left(\frac{1}{2}\right)\sum_{q=1}^{n_q}\left(U_q + dU_q\right)^T R_q\left(U_q + dU_q\right) \\ + \lambda^T\left(dY_N - \sum_{q=1}^{n_q} B_q dU_q\right)\end{aligned} \tag{33}$$

$\lambda \in \Re^n$ is Lagrange multiplier (adjoint variable). Necessary conditions of optimality for $q = 1, 2, \ldots, n_q$ are as,

$$\frac{\partial \bar{J}}{\partial dU_q} = R_q\left(U_q + dU_q\right) - B_q^T\lambda = 0 \tag{34}$$

$$\frac{\partial \bar{J}}{\partial \lambda} = dY_N - \sum_{q=1}^{n_q} B_q dU_q = 0 \tag{35}$$

Here, B_q is function of states ((27)), however, dU_q is considered as independent decision variable and not function of state or control at previous time step. From (34),

$$dU_q = R_q^{-1}B_q^T\lambda - U_q \tag{36}$$

substituting dU_q from (36) in (35), we get,

$$\lambda = A_\lambda^{-1}\left(dY_N + b_\lambda\right) \tag{37}$$

here, $A_\lambda = \left[\sum\limits_{q=1}^{n_q} B_q R_q^{-1} B_q^T\right]$, $b_\lambda = \sum\limits_{q=1}^{n_q} B_q U_q$ and it is assumed that A_λ is nonsingular. Substituting (37) in (34), gives,

$$dU_q = R_q^{-1}B_q^T A_\lambda^{-1}\left(dY_N + b_\lambda\right) - U_q \tag{38}$$

updated control is,

$$U_q + dU_q = R_q^{-1}B_q^T A_\lambda^{-1}\left(dY_N + b_\lambda\right) \tag{39}$$

for $q = 1, 2, \ldots, n_q$
Steps of MPSP for impulsive control system are summarized as follows:
1) Guess initial control history $U_1, U_2, \ldots, U_q$ at impulse instants, $t_1, t_2, \ldots, t_q$ and propagate system using (6), (7) and (10).
2) Compute dY_N using (10) and check for convergence criteria of dY_N $\left(Y^d - Y_N\right)$ i.e., $dY_N \approx 0$ or any other suitable criteria such as l_2 norm $((\|dY_N\| / \|Y_N\|) \approx 0)$ which will ensure $Y_N \to Y^d$. Note that intial guess control history may not satisfy convergence criteria but it should ensure Y_N is close to Y^d.
3) If convergence criteria is not satisfied then compute sensitivity matrices recursively using (28)- (30) and update control history by using (39).
4) steps 1) is followed with updated control history as new guess control history and convergence criteria is checked by step 2).
5) If convergence criteria is met, then algorithm is stopped otherwise step3) is followed.

3. MOTIVATING EXAMPLE: IMPULSE CONTROL FOR PREDATOR-PREY (LOTKA-VOLTERRA) SYSTEM

3.1 System Dynamics

Predator prey (Lotka Volterra) model with impulsive control is considered. Model is chosen from Li et al. [2009] and parameters with units are taken from weblink (http://www.stanford.edu/∼fringer/teaching /numerical_methods_02/ assignments/assignment3.pdf). Predator prey is population interaction model between predator and prey. It is described as follows.

$$\dot{x}_1 = x_1\left(\mu_1 - r_{12}x_2\right) \tag{40}$$

$$\dot{x}_2 = x_2\left(-\mu_2 + r_{21}x_1\right) \tag{41}$$

$$x_{1k}^+ = x_{1k}^- + u_{1q} \tag{42}$$

$$x_{2k}^+ = x_{2k}^- + u_{2q} \tag{43}$$

here, $q = 1, 2, ..., n_q$ and n_q is maximum number of impulses. (42) and (43) at applicable at impulse instants, otherwise (40) and (41) are applicable. x_1 represents prey, x_2, represents predator. Model assumptions are:: i) prey has enough food for its survival ; ii) prey population is decreased after its interaction with predator otherwise prey will grow exponentially; iii) predator is surviving only on prey and it has no other source of food consumption; iv) predator prey interaction lead to increase in predator population; v) predator population has natural death rate which will decrease its population exponentially. From (40), μ_1 is growth rate of prey, r_{12} is interaction constant which results in decrease of prey, μ_2 is death rate of predator and r_{21} is interaction constant which results in increase of predator. Impulsive effect on predator prey model is described in (42) and (43). Physically, impulsive effect can be due to force eradication of predator and prey or due to emigration or some natural cause. u_1 and u_2 represents impulse control. Negative value of u_1 and u_2 represents harvesting of fish and sharks respectively, positive value of u_1 and u_2 represents putting in of fish and sharks respectively (fish and sharks can be available in stocks).

3.2 Control design

Aim of the control design is to drive system states towards its equilibrium point. Parameter values of model ((40)-(43)) are $\mu_1 = 0.7$ per year, $\mu_2 = 0.5$ per year, $r_{12} = 0.007$ per year and $r_{21} = 0.007$ per year. One of the equlibrium points of the system ((40)-(41)) is $x_1^* = \mu_2/r_{21} = 71.4$ and $x_2^* = \mu_1/r_{12} = 100$. As fish and shark cannot be in fractions, equilibrium point are taken as $(x_1^*, x_2^*) = (72, 100)$. Thus, desired values of states are, $x_1^d = 72$ and $x_2^d = 100$. Time duration of control is taken as one year with nine impulses (ten equal interval between inital and final time), whereas plots are shown upto two years. Time interval, ΔI, between impulse is chosen as $\Delta I = 0.1$ year. Thus, total impulses n_q is, $n_q = (t_f/\Delta I) - 1 = 9$. Hence, total number of segments will be, $n_{seg} = n_q + 1 = 10$. Total time interval is divided in N distinct time instants, $N = ((t_f - t_0)/dt) + 1$, dt is step size for RK4 integration, $dt = 0.01$ thus, $N = 101$. As, n_q, n_{seg} and N are known, n_d is found using $N +$

$n_q = n_d n_{seg}$, thus, $n_d = 11$. Error criteria for Case I is $\|dY_N\|_2 < 0.0001$, where, $\|dY_N\|_2$ is l_2 norm of dY_N and is given by, $\|dY_N\|_2 = \left[\left(x_1^d - x_{1N}\right)^2 + \left(x_2^d - x_{1N}\right)^2\right]^{(1/2)}$ and for Case II is $\|dY_N\|_2 < 0.0001$, where dY_N is given by, $dY_N = x_1^d - x_{1N}$.

3.3 Simulation Results

Initial guess control history guess is taken as zero. Fig.2 shows the variation of states x_1 (fish) and x_2 (sharks) after application of impulse control. Initial conditions are $x_{10} = 1000$ fish and $x_{20} = 50$ sharks. It is observed that x_1 and x_2 go close to its equilibrium points after ninth impulse. Even though fish population has positive growth rate, it is decreasing because of harvesting and shark population. It is also expected because initial condition of fish population is above the equilibrium conditions. Shark population is increasing from its initial conditions towards equilibrium. Fig.3 shows the impulse control i.e., harvesting strategy for fish and shark so that equilibrium point is reached in one year.

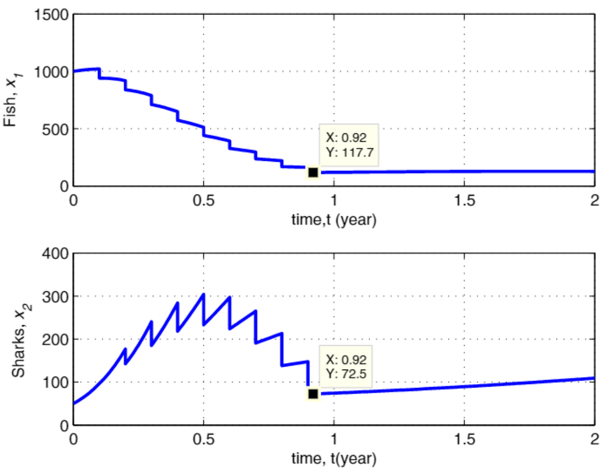

Fig. 2. Prey and predator population after application of impulse control

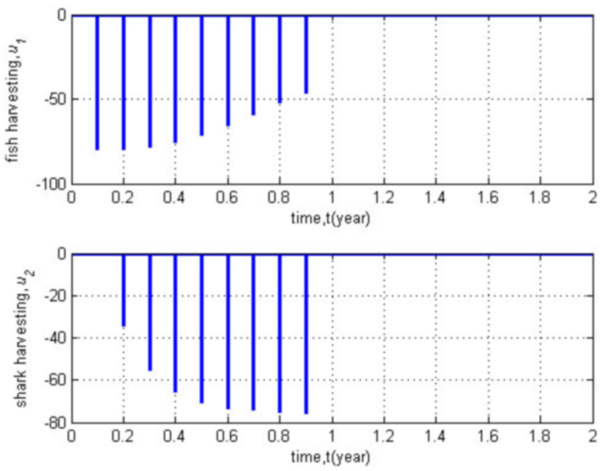

Fig. 3. Impulse control applied at an interval of 0.1 year.

4. CONCLUSIONS

A new computationally efficient algorithm for impulsive control system is proposed. It is extension of model predictive static programming for continuous control systems. Simulation results for predator prey model are shown. This algorithm is fast because of recursive computation of sensitivity matrices and control is computed only at impulse instants (not at all time steps). For, predator-prey system it takes 1-2 sec to run complete program. Algorithm demands good initial control history guess, but from our past experience (model predictive static programming for continuous systems) it is found that most of the problem get solved even when initial guess of control history is taken as zero. Interval between impulses can be fixed at equal or unequal intervals. There can be difficulty in finding solution when all or some of the intervals between impulses are large or total number of impulses are not adequate to give solution. Absolute value of summation of optimal control values decreases when number of impulses are increased by keeping fixed final time.

REFERENCES

J. T. Betts. *Practical Methods for Optimal Control using Nonlinear Programming.* Society for Industrial and Applied Mathematics (SIAM), 2001.

S. H. Hou and K. H. Wong. Optimal impulsive control problem with application to human immunodeficiency virus treatment. *Journal of Optimization Theory and Applications*, 151:385–401, 2011.

D. Kemih and M. Chahim. Numerical algorithms for deterministic impulse control models with applications. *Discussion Paper from Tilburg University, Center for Economic Research, October, 2012.*, 2012.

D. E. Kirk. *Optimal Control Theory: An Introduction.* Dover Publications, 2004.

D. Li, S. Wang, X. Zhang, and D. Yang. Impulsive control of uncertain lotka-volterra predator-prey system. *Chaos, Solitons and Fractals*, 41:1572–1577, 2009.

R. Padhi and M. Kothari. Model predictive static programming: a computationally efficient technique for suboptimal control design. *International Journal of Innovative Computing, Information and Control*, 05:399–411, 2009.

S. S. Rao. *Engineering Optimization: Theory and Practice.* Wiley, 2009.

P. J. Werbos. *Approximate dynamic programming for real-time control and neural modeling.* Handbook of Intelligent Control, D. A. White and D. A. Sofge (eds.), Multi-Science Press, pp. 493-526, 1992, 1992.

X. Wong and S. N. Balakrishnan. Optimal neurocontroller synthesis for impulse-driven systems. *Neural Networks*, 23:125–134, 2010.

H. Yang, L. Dou, and M. Gan. A particle swarm optimization for fuel-optimal impulsive control problems of guided projectiles. *22nd Chinese Control and Decision Conference, Xuzhou, China, May 26-28, 2010*, 2010.

T. Yang. *Impulsive Control Theory.* Springer, Berlin Heidelberg, 2001.