An Efficient Finite-Input Receding Horizon Control Method and Its Application for the Pneumatic Hopping Robot

Jian Wu
Engineering Department, Omron Oilfield and Marine, Houston, TX 77088, USA
Email: jian.wu@omron.com

Sherif Abdelwahed
Electrical and Computer Engineering Department, Mississippi State University, USA
Email: sherif@ece.msstate.edu

***Abstract*— In this paper, a finite-input receding horizon controller (FIRHC) is proposed as motivated by the need to use solenoid valves to control the motion of a pneumatic hoping robot. The controller aims to the application on switching control systems in which only a finite number of control inputs are available. The controller utilizes a model to predict system behavior along a finite forward horizon, and establishes an optimization problem, and then finds an optimal control sequence that gives the optimal cost and ultimately only the first element of the sequence is applied at each time step. The stability issue of the controller is discussed as a terminal equality constraint is added. Since only finite discrete inputs exist, the analytical solution is usually not possible to achieve, and exhaustive search was generally the approach to get the optimal control input. As is known, the exhaustive search becomes computationally prohibitive with an increasingly long horizon. An efficient modified depth first search algorithm is proposed, namely, sorted depth first search (sDFS). It preserves the completeness of exhaustive search, while significantly reducing time and space complexity. The whole approach is applied to a pneumatic hopping robot system where the motion control is re-formulated as an explicit energy regulation problem. The control goal is to maintain the system energy at a desired level. An additional example on a three tank control system is used to further illustrate the efficiency of sDFS method on the system with possession of a relatively large amount of modes. Simulation results demonstrate the effectiveness of the proposed method.**

***Index Terms*—receding horizon control, switching systems, finite control set, sorted depth first search, energy regulation, hopping robot**

I. INTRODUCTION

In this paper, we introduce a feasible and efficient control strategy to regulate the oscillatory motion of a pneumatic hopping robot. Because pneumatic actuators can provide much higher power-to-weight ratio than their electrically actuated competitors, they have been used directly as legs to drive a legged-robot since late 1980s [1]. In the work of Binnard [2], a small six-legged pneumatic walking robot was designed using customized lightweight pneumatic actuators. The performance clearly showed high force and power density, which means that the robot can walk faster with larger payloads. Other advantages of using pneumatic systems such as energy storage, and natural compliance for shock absorption, provide latitude for stable and energy efficient controller design with pneumatically actuated systems. The absorbed energy can be stored as the internal energy of compressed air and can be released again when the hopper is in flight. Fast gaits are generated for the control of a pneumatically actuated robot in [3]. The control system generates the desired trajectories on line and generates proper control inputs to achieve the desired trajectories. An energy-based Lyapunov function was chosen to generate the controlled limit cycles. This is similar to the work in [4], except that a desired velocity based on the current position and direction of motion is generated based on energy conditions to be the reference trajectory for the control system to track.

As discovered in practice, the use of proportional valves presents a costly and bulky option for a system that is ideally lean and inexpensive. For example, in order to realize the hopping control for a multiple-legged robot, the number of valves needed quickly becomes prohibitive in terms of both cost and size. Therefore, low-cost solenoid valves were used to exploit the hopping control approaches instead. In [4], though the developed control method demonstrated its effectiveness, it has own limits such as, the position-based velocity and acceleration rate reference trajectories are complicated to generate; and finding the reference value (look-up table) and getting derivative of acceleration rate for computing the control input are difficult. Therefore, we are motivated to investigate a simpler and more straightforward control strategy to realize the desired hopping control.

Receding horizon control (RHC) or model predictive control (MPC) theory has adequately proved its usefulness in practice over the past decades [5, 6, 7, 8] (and the reference therein). This general approach describes a class of control methods, which make explicit use of process models to predict future system behavior over a limited time horizon, compare this behavior to an expected behavior using a pre-specified objective function, and then generate a control signal sequence that optimizes the objective function over the limited time horizon. The first signal in the sequence is chosen for the current step. This procedure is repeated to generate control signals for the plant in a step by step manner. This methodology offering attractive features, such as explicit use of mathematical models, and handling the system constraints, is often taken as a practical realization of optimal control, considering the fact that the main difference is optimal control uses an infinite horizon which usually is not tractable for the real time implementation.

We categorize the on/off three-way valves pneumatic hopping robot control system into a generalized kind of dynamic system, namely, switching control system. Switching control system represents a type of dynamic systems in which control inputs are discrete –valued and selected from a finite set. Here we aim at realizing the pneumatic hopping control, more generally switching control, by combining with the industry-proven receding horizon control strategy. In one of original work in this area, Tsang and Clarke [9] defined the case as an optimal bang-bang predictive control problem and uses exhaustive search to derive the solution at each time step. Since the corresponding number of permutations is the number of discrete control values raised to a power corresponding to the number of prediction horizon, such a proposed exhaustive search can quickly become computationally prohibitive. Later on, Abdelwahed, and Wu, et al. [10, 11] extended similar strategy into a class of hybrid systems (i.e. mixed continuous and discrete system) where exhaustive search is still taken as the solver tool for the formulated optimization problem. Shields, Barth, and Goldfarb [12] proposed a method for the predictive control of switching systems characterized by linear system dynamics with a time delay in the input channel. The proposed controller predicts the states of the system at an appropriate future time horizon for each of the present-time discrete-valued input candidates. The error between the predicted states and the future desired states is calculated for each of the inputs and the input candidate that produces the minimum predicted Lyapunov function is chosen. It is shown to have a bounded tracking error for stable plants. The method is restricted to linear systems and requires measurement of the full state, and it uses the simplest case of predictive control to afford the computational cost where only single step forward horizon is employed.

In this work, we propose a receding horizon control method to regulate switching actions of the solenoid valves for hopping control, where particularly a terminal constraint is added to ensure the stability of the system. We first establish a mapping from a desired motion specification to a desired total conservative energy.

Therefore the control objective turns to be regulating the conservative energy of system. As found that in the hopping system, energy loss (small friction and leak) is usually insignificant, namely, the energy varies within a relatively small region of desired energy value, being suitable to the inconvenient limitation that the terminal constraint restricts the operational region. And the predictive control problem is still tractable to solve using a reasonable forward horizon. In additions, we present a more efficient solver approach to find the control sequence. The basic idea is motivated from the fact that cost function values are always positive along any forward horizon, and then if at the early stage of the search low cost values can be established to be "threshold", many searches can be in advance terminated without affecting completeness if the accumulated cost along a control sequence is above the "threshold" even the search has not reached the end of the horizon. And the subsequent searching effort along the sequence with different permutations thereafter becomes unnecessary and therefore can be avoided and the relevant computation cost is therefore saved.

The paper is organized as below. In Section II, the finite-input receding horizon controller design is presented, and a general searching algorithm for reducing computation effort is proposed. Section III describes a pneumatic vertical hopping robot system, and shows the procedures to derive the desired total conservative energy value from the given desired hopping period specification. In Section IV, simulation results are given to demonstrate the effectiveness of the proposed control method and the searching algorithm. Section V concludes the paper.

II. FINITE-INPUT RECEDING HORIZON CONTROLLER DESIGN

II. 1. Finite-Input Receding Horizon Control Law

The following discrete form of state equations is used to describe the behavior of a switching control system:

$$x(k+1) = f(x(k), u(k)), \tag{1}$$

where $x(k), x(k+1) \in R^n$, and $u(k) \in S$. S represents a finite control set.

The main idea of receding horizon control is to use a model of the plant to predict the future evolution of the system. Based on this prediction, at each step k, the controller select a best sequence of control inputs by solving an optimization problem, which minimizes the tracking error, and in the meanwhile satisfies the active system constraints. Once the best sequence is obtained, only the first element of the sequence is actually applied to the plant. At next time step k+1, this procedure is repeated.

Consider the following optimal control problem

$$\min_{u \in U} J = \sum_{i=k}^{k+N-1} \lambda \| x\ (i) - x_s \|^2 + \beta \| u(i) - u_e \|^2, \tag{2}$$

such that

$$\begin{aligned} & x(k+N) = x_e, \\ & x(k+1 \mid k) = f(x(k), u(k)), \\ & u_{\min} \le u(k) \le u_{\max}, \end{aligned} \tag{3}$$

where (x_e, u_e) is an equilibrium pair. (Definition 1 – Equilibrium pair: A vector $x_e \in R^n$ and input $u_e \in S$ are said to be a equilibrium pair for system (1) if $f(x_e, u_e) = x_e$, for $\forall t > 0$.) N is the prediction horizon, λ is a vector representing the relative importance (preference) of each variable in the cost function, and β is a vector penalizing control effort which aims to prohibiting frequent switching control. $x(k)$ and $u(k)$ are subject to the constraints (3). Assume for the current time instant k, the optimal solution U_k^* = { $u^*(k)$, $u^*(k+1), \ldots, u^*(k+N-1)$ } exists. According to the receding horizon philosophy, the applied control action is set

$$u(k) = U_k^*(1), \tag{4}$$

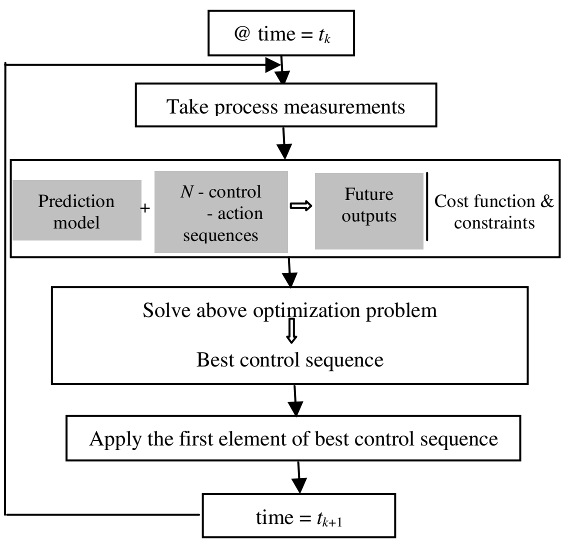

Figure 1. RHC using finite discrete inputs – FIRHC law

while the subsequent optimal inputs in U_k^* are disregarded. At next step, this whole optimization procedure is repeated. The control law (2) - (4) is referred as the finite-input receding horizon control (FIRHC) law. Figure 1 shows this general idea of this control law.

Theorem 1: Let (x_e, u_e) be an equilibrium pair[2]. Providing the optimization problem (2) is feasible and is solved at each step, the FIRHC controller (2) – (4) stabilizes the system.

Proof: The proof follows from Lyapunov arguments. Let U_k^* be the optimal control sequence at time instant k in which U_k^* = { $u^*(k), u^*(k+1), \ldots, u^*(k+N-1)$ } and N is a pre-specified forward horizon. And let $\tilde{U}$ be a control sequence { $u^*(k+1)$, $u^*(k+2), \ldots, u^*(k+N-1), u_e$ }. Consider that U_k^* is such a control sequence that leads to x_e, then $\tilde{U}$ is feasible at time k+1. Hence,

$$\begin{aligned} J_{opt}(k+1) &\le J_{\tilde{U}}(k+1) \\ &= J_{opt}(k+1) - \| x(k) - x_e \| - \| u(k) - u_e \|, \end{aligned} \tag{5}$$

and thus $J_{opt}(k)$ is decreasing. $J_{opt}(k)$ is also lower-bounded by 0. Therefore, the theorem is proved.

Remark 1: The terminal equality constraint in (3) is added to ensure the stability of RHC approaches. The similar way to ensure the stability has been studied in the other linear and nonlinear RHC cases [5, 6]. With this constraint, the state is at x_e at the end of the finite horizon and therefore the control action is u_e; consequently if there are no disturbances and a perfect prediction model is used, the system stays at the equilibrium state.

Remark 2: However, the terminal constraint added also gives rise to more computational burden, and generally leads to a restrictive operating region, e.g., a region near the equilibrium state. In this regard, the theorem is rather about local stability than global stability. As is normally the case, Theorem 1 gives the sufficient condition for the stability.

Remark 3: The theorem provided assumes the feasibility to imply stability. The added equality constraint adds the difficulty to achieve the feasible solution. It is clear that short horizons are desirable from a computational point of view. However, long horizon is necessary, at least helpful, to achieve the feasible solutions to ensure stability and enhance the closed loop performance. At next, a general approach is proposed to aiming at reducing the computational cost for seeking the solution.

II. 2. Sorted Depth First Search Algorithm

Given the fact that the total number of control inputs is finite and the forward prediction horizon is finite, the number of candidate control sequences as the solution for the optimization problem mentioned above is finite as well. Therefore, to solve the problem can be cast as a search process. As shown in [9, 10,11], a general approach to solve the discrete input involved predictive control problem is to use exhaustive search method in which all the permutations of control inputs sequence over the prediction horizon are exhaustively searched. Since the corresponding number of permutations is the cardinality of the control set raised to a power of the number of prediction horizon. Such a proposed exhaustive search can rapidly become computationally prohibitive as the horizon is defined longer.

We propose a general algorithm to sufficiently improve the computational efficiency. The motivation is taking advantage of the fact that all cost values are positive that are summed up along the forward horizon. The basic idea is that, if a low control sequence cost is found at the early stage in the searching, then many subsequent candidate sequences can be cut off even before they reach at the end of the horizon, providing that the predicted cost at the middle of sequence already exceeds that low cost value. As a result of "cut off", all the following permutations of that sequence can be cut off as well. Therefore, a lot of unnecessary computations will be avoided.

To implement the idea, a notion of node is defined. A node is a vector that is composed of the states, control action, depth and accumulated cost. The search is starting from the rooting node (i.e., current state) and is going further and further until reaching the end of horizon (similar to depth first search [13]), while expanding the new nodes along the horizon using state equations. Particularly, all freshly expanded nodes at the same depth, named as peers, will be sorted first by the cost value and added to a last-in-first-out stack for later expansion. The node with the lowest cost value among the peers will be put to the last of stack, which means, the node with lower cost value will get expanded first. As such, once the first sequence cost is achieved (expectedly a mildly low value), this value is stored as "optimal cost" which can be updated till the end of search. From then on, only the node with smaller cost than "optimal cost" can be added to the stack. As the searching along a sequence reaches the end of horizon (i.e., depth first), the search backtracks, returning to the most recent (last) node it has not finished exploring in the stack. The searching process terminates until the stack becomes empty.

In the regular cases, the space complexity of this algorithm is much lower than the exhaustive search since it only stores those nodes that are useful and qualified to expand. The time complexity is proportional to the number of total nodes once added to the stack that has been sufficiently reduced by the sorting process as mentioned above. The algorithm is presented in Table 1. In Sections IV, simulation results are provided to demonstrate the efficiency of the algorithm.

III System Description of a Pneumatic Hopping Robot

III.1 Description for the Hopping Motion

This work considers a vertical, gravity influenced pneumatic piston carrying an inertial load, serving as a hopping robot, as shown in Figure 2. The system contains two control inputs in the form of on/off three way valves that influence the mass flow into or out of each chamber *a* and *b*.

The kinetic and potential energy terms for a leakless, adiabatic (no heat losses), frictionless piston-mass system while in contact with the ground are given as [4]:

$$E = \underbrace{\frac{1}{2}M\dot{x}^2}_{KE(\dot{x})} + \underbrace{\frac{P_aV_a}{1-\gamma}\left[\left(\frac{P_a}{P_{atm}}\right)^{\frac{1-\gamma}{\gamma}} - 1\right]}_{PEa(x)} + \underbrace{\frac{P_bV_b}{1-\gamma}\left[\left(\frac{P_b}{P_{atm}}\right)^{\frac{1-\gamma}{\gamma}} - 1\right]}_{PEb(x)} + \underbrace{P_{atm}A_r x}_{PEr(x)} + Mgx \quad (6)$$

TABLE 1. SORTED DEPTH FIRST SEARCH ALGORITHM

```
---------------------------------------------------------------------
Struct node {states, mode, depth, cost};
Stack node_array, temp_node_array;
Struct node current_node, new_node;

Array path, best_path;
Const horizon, mode_no;
Var mode, best_cost;

best_cost = +∞ ; //assign the initial cost
Push (node_array, root_node); //initialize with the current state

WHILE node_array ~= empty
 current_node := Pop(node_array); //remove last element of stack
 path(current_node.depth):=current_node.mode;  //remember current
   path

 IF current_node.cost < best_cost THEN
   FOR i =1: mode_no
       SWITCH {mode}
              CASE(mode)
                    new_node := Compute(current_node, mode);
                    //a function to generate a child node, depth++
        ENDSWITCH

        IF new_node.cost < best_cost && new_node.depth ~=
        horizon THEN
           Push(temp_node_array, new_node);
        ENDIF

        IF new_node.cost < best_cost && new_node.depth ==
        horizon && Terminal_equality_check( ) == true THEN
           best_cost := new_node.cost; //update best_cost
           path(new_node.depth) := mode;
           best_path := path; //update the best path so far
        ENDIF
   ENDFOR

   IF temp_node_list is not empty THEN
      Sort_descending(temp_node_list); //sort the nodes by costs
      Push(node_array, temp_node_list);
      //push in last the node with lowest cost
   ENDIF

 ENDIF
ENDWHILE

u* := best_path(1); //the first element of the best path is the one wanted
---------------------------------------------------------------------
```

where $V_{a,b}$ represents the volume of chamber a or b, and $A_r = A_a - A_b$ represents the cross sectional area of the piston rod. The potential energy of each chamber of the actuator is derived using standard thermodynamic relationships as the ability of the pressure in the chamber, P_a or P_b, to do work adiabatically with respect to an environment at atmospheric pressure P_{atm}, where the ratio of specific heats is denoted by γ. The term, $PE_r(x)$ is a term similar to a gravitational potential energy term due to the unequal piston areas of the two sides of the actuator. If the system has no losses, the system will maintain a constant energy E by shuttling energy between potential and kinetic energy storage in the form of a well defined oscillation.

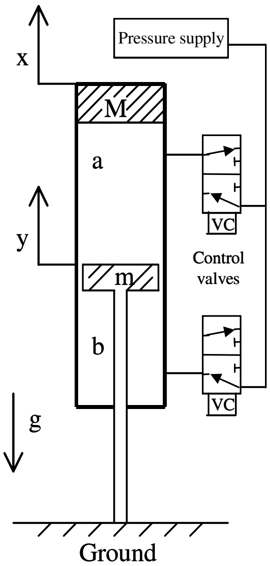

Figure 2. Schematic of a pneumatic robot

Ideally (no energetic loss), the motion equations can be obtained by setting $\dot{E}=0$:

$$M\ddot{x} = P_a A_a - P_b A_b - P_{atm} A_r - Mg \quad (7)$$

$$m\ddot{y} = P_b A_b - P_a A_a + P_{atm} A_r + F_{ground} - mg \quad (8)$$

Expressions for the time rates of change of the pressures can be derived from the following constitutive relations for the rate of internal energy storage $\dot{U}$, rate of heat input $\dot{Q}$, enthalpy rate $\dot{H}$, and work rate $\dot{W}$, of each control volume associated with side a and side b:

$$\dot{U} = \dot{Q} + \dot{H} - \dot{W} \quad (9)$$

$$\dot{U} = \frac{\dot{P}V + P\dot{V}}{\gamma - 1} \quad (10)$$

$$\dot{H} = \dot{m} c_p T_{flow} \quad (11)$$

$$\dot{W} = P\dot{V} \quad (12)$$

These result in the following for $\dot{Q}=0$ (adiabatic):

$$\dot{P}_a = \frac{\gamma R T_{flow}}{V_a}\dot{m}_a - \frac{\gamma P_a \dot{V}_a}{V_a} \quad (13)$$

$$\dot{P}_b = \frac{\gamma R T_{flow}}{V_b}\dot{m}_b - \frac{\gamma P_b \dot{V}_b}{V_b} \quad (14)$$

The standard equations for the mass flow through the orifice are given by Ben-Dov and Salcudean [14]:

$$\text{If } \frac{P_d}{P_u} \le 0.528, \text{ then } \dot{m}_{a,b} = C_f A_v C_1 \frac{P_u}{\sqrt{T}},$$

otherwise, (chocked flow)

$$\dot{m}_{a,b} = C_f A_v C_1 \frac{P_u}{\sqrt{T}} \left(\frac{P_d}{P_u}\right)^{1/k} \sqrt{1 - \left(\frac{P_d}{P_u}\right)^{(k-1)/k}}. \quad (15)$$

where A_v is orifice area, C_f is discharge coefficient, T is temperature of flow, P_u and P_d are, respectively, upstream and downstream pressure.

A hopping cycle can be defined as shown in Figure 3. The corresponding moments and periods are defined [4]:

$t_0 = 0$ defines the starting point of a full hopping cycle ($x = 0$).

t_1 defines the lift-up moment of the foot ($y = 0$).

$T_1 = t_1 - t_0$ is defined as the launch period.

T_2 is defined as the compression period ($x \le 0$).

T_{air} is defined as the flight time.

T_3 is defined as the recovery period.

T_{hop} is defined as the full hopping period.

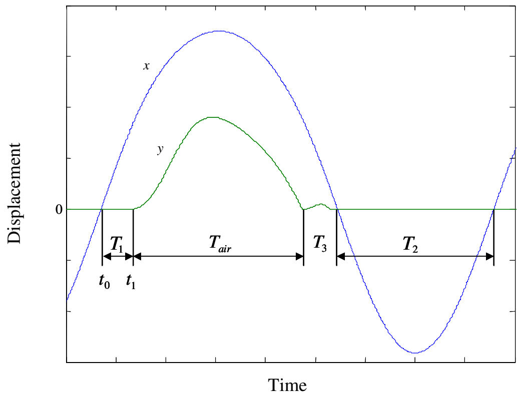

Figure 3. A hopping cycle

As defined above, the hopping cycle can be analyzed according to the following decomposition,

$$T_{hop} \cong T_1 + T_3 + T_{air} + T_2 \quad (16)$$

where T_{hop} and T_{air} are to be specified, and $T_3 \cong T_1$.

III.2 Derivation of the Desired Total Conservative Energy

It should be noted that the conservative energy storage expression of Equation (6) is valid only when the contact with ground is maintained. It is sufficient to control the energy storage during contact only. Therefore, the proposed control strategy is to switch the valves only in the contact period to regulate the total energy around the desired point (thus the desired hopping motion can be achieved), and seal off the valves in flight. A critical step is to achieve the desired conservative energy value given the pre-specified motion index - total hopping period T_{hop} and flight time T_{air}.

The procedure to derive the desired total conservative energy of the system to support the desired T_{hop} and T_{air} is summarized below. For the limit of space, only critical equations are listed. For more details, find [4].

Given T_{hop} and T_{air}, compute T_2 by solving the following equation:

$$T_{hop} \cong \frac{2T_2}{\pi} \arcsin\left(\frac{8(M+m)g}{8(M+m)g + M \cdot (\pi / T_2)^2 g T_{air}^2}\right) + T_{air} + T_2 \quad (17)$$

Compute $x(t_1)$ and $\dot{x}(t_1)$ using the following equation and results from (17)

$$x(t_1) \cong \frac{(M+m)g}{M \cdot (\pi / T_2)^2} \tag{18}$$

$$\dot{x}(t_1) = \frac{T_{air} g}{2} \tag{19}$$

Utilizing the above results and substituting into Equation (6), compute the desired total conservative energy of the system:

$$E_d = \tfrac{1}{2} M\dot{x}(t_1)^2 + PEa(x(t_1)) + PEb(x(t_1)) + P_{atm} A_r x(t_1) + Mgx(t_1) \tag{20}$$

Since it is known that the energy of the system when leaving and touching with the ground is identical (except for losses), regulating the system to maintain the desired total energy (ideally, hopping with constant energy) in contact will result in the desired hopping motion.

IV SIMULATION EXPERIMENTS

IV.1Experiment 1

A simulation of a controlled pneumatic hopping robot involving frictional losses is presented as below. The hopping system was modeled as the following with added friction terms of equations (7) and (8):

$$M\ddot{x} = P_a A_a - P_b A_b - P_{atm} A_r - Mg - b(\dot{x} - \dot{y}) \tag{21}$$

$$m\ddot{y} = P_b A_b - P_a A_a + P_{atm} A_r + F_{ground} - mg - b(\dot{y} - \dot{x}) \tag{22}$$

with M = 0.54 kg and m = 0.05 kg, and viscous friction effects representing the sliding piston and rod seals modeled by b = 2 Ns/m. The pressure dynamics were modeled as equations (13) and (14). The ground model was approximated as a very stiff spring and damping to represent losses upon collision,

$$F_{ground} = -k_{ground}\, y - b_{ground}\, \dot{y}, \quad \text{if } y < 0, \tag{23}$$

where k_{ground} = 1.0×10^6 N/m and b_{ground} = 1000 N.sec/m.

The procedure listed in Section III.2 is used to derive the desired total conservative energy given a pre-specified motion index. For example, given T_{hop} = 0.4s and T_{air} = 0.2s, T_2 can be pre-computed by solving equation (17) using a nonlinear solver. As a result, T_2 = 0.161s. Following the procedure, the desired energy E_d = 11.44. In this case, the control of side a is to compensate mostly for leakage, the mass flow rate must be greater than some number when the chamber is under pressurized, the following on/off control law will be used for side a while in contact with the ground: Charge chamber a if $P_a < P_{a0}\left(\frac{V_{mida}}{V_{mida} + A_a x}\right)^{\gamma}$ and $y \le 0$, where P_{a0} and V_{mida} are respectively static equilibrium pressure and volume of chamber a. The mass flow into chamber b is used to maintain the energy. Supply pressure P_s = 80 psi. Here only the terminal equality constraint on total energy is enforced. As the desired velocity reference used is approximately ideal with no friction and other losses, it may make the problem intractable to require a terminal constraint on the velocity term. The other setting for (2) is that $\lambda_1 = 1, \lambda_2 = 0.1$, and $\beta = 10$. Horizon N is set to be 6. For this problem, Equation (2) and (3) becomes

$$\begin{aligned}
&\min_{u \in U} J = \sum_{k}^{k+5} (E(i+1) - E_d)^2 + 0.1(\dot{x}(i+1) - \dot{x}_d(i+1))^2 + 10u(i)^2 \\
&\text{s.t.} \\
&E(i+1) \text{ is subject to Eqn. (6), (7), (8), (13), (14), and (15),} \\
&E(k+6) = E_d, \\
&-0.01 \le u(i) \le 0.01.
\end{aligned} \tag{24}$$

Fig. 4 – 8 show the simulation results of the hopping robot for T_{air} = 0.4s and T_{air} = 0.2s. Fig.4 shows the hopping behavior. As a result, the average T_{hop} ≅ 0.39 and T_{air} ≅ 0.19s that means the motion is well controlled. It is better than previous work [4] where T_{hop} ≅ 0.39 and average T_{air} ≅ 0.18s. Fig.5 shows that the system is stabilized in a desired circle. Fig.6 displays the pressures' time evolution. Fig.7 and 8 give close looks on the total energy evolution and mass flow b during a compression period. They demonstrate under the unpredicted friction, the proposed controller effectively regulates system energy around the desired point.

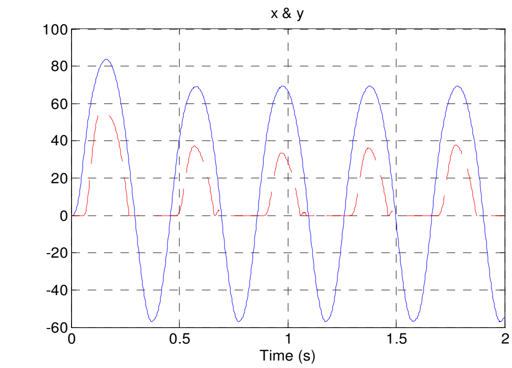

Figure 4. Displacements of hopping robot, dotted: y

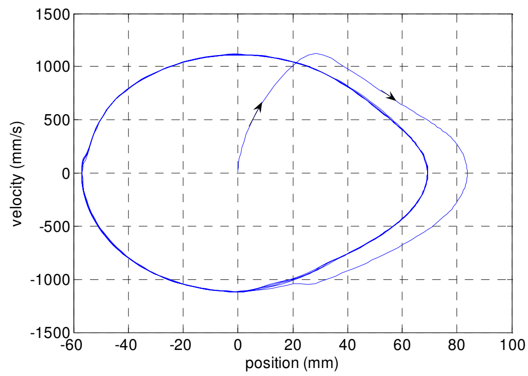

Figure 5. Position vs. velocity

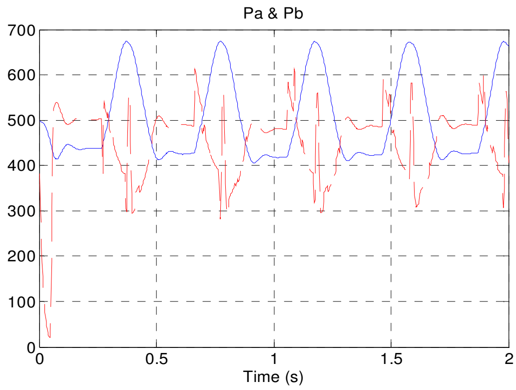

Figure 6: Pressure of Chamber *a* and *b*, dotted: Chamber *b*

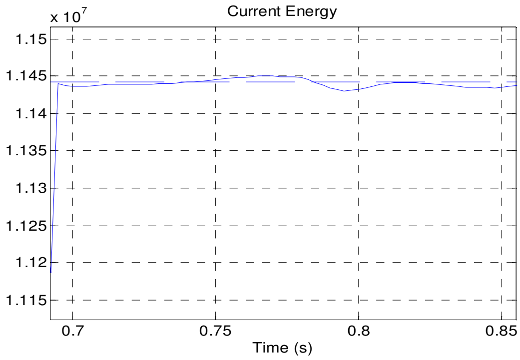

Figure 7: Energy regulation in a compression period ($T_2 = 0.165$), dotted: desired energy

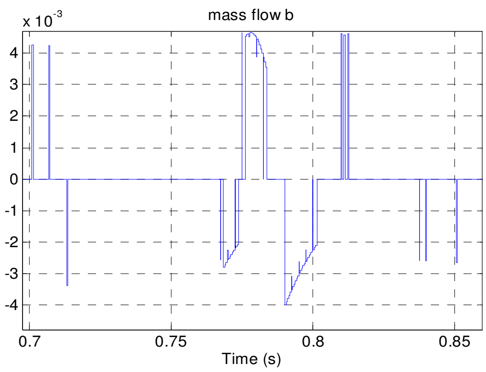

Figure 8: Mass flow into Chamber b in a compression period

Several sets of 2-second-simulations are conducted to check on the efficiency of the proposed sDFS method. Table 2 displays the results for cases of horizon = 5, 6, 7, and 8. Note that Point A represents nodes visited at a step in ES, Point B represents nodes visited at a step in sDFS, (max/min), and Point C denotes the total time of sDFS /total time of ES (%).

The final column (Point C) is achieved by executing two methods on the same experiment settings of the hopping system. It is apparent that sDFS always has fewer nodes to visit by comparing the elements between Point B column and Point A column. And as horizon becomes longer, the sDFS has better and better performance compared to the exhaustive search (ES).

TABLE 2: EXHAUSTIVE SEARCH AND SORTED DEPTH FIRST SEARCH FOR EXPERIMENT 1

	Point A	Point B	Point C
5	363	363/15	7.0%
6	1,092	1,044/18	4.1%
7	3,279	2,208/21	2.3%
8	9,840	4,380/24	1.1%

IV.2 Experiment 2

To further illustrate the efficiency of sDFS algorithm when it is applied to a relatively large multi-mode system, a three tank fluid system example is introduced. The fluid system consists of three identical tanks, connected in a series configuration. Transfer of fluid through different paths is made possible by on/off control of a set of solenoid operated valves in the system. The fluid flow into and out of each tank is controlled by three valves: (i) an inflow, (ii) an outflow, and (iii) a bypass valve. Three additional valves in the system provide further control of the fluid flow in the system. The fluid level in each tank is measured. This system has been successfully realized online using the similar control strategy [10]. Intending to simplify the computations, the total operation modes that can be up to $2^7 = 128$ have been reduced to 17 in which the controllability is well maintained. The prediction horizon still has to be limited to no larger than four since only exhausted search was used then to derive the best control solution at each step.

Given a particular mode of system operation – The second tank is filling at low speed and the first tank transferring fluid to the third tank. For this mode, $u = \{V_{Fill1} = 0,\ V_{Fill2} = 1,\ V_{trans1} = 1,\ V_{trans2} = 0,\ V_{trans3} = 1,\ pump = \text{low}\}^T$, and $Flow_{low}$ is the corresponding constant flow provided by the pump with low speed. The discrete time state equations for this mode are

$$\begin{aligned} h_1(k+1) &= h_1(k) + t_s((h_3(k) - h_1(k))/\tau_1(k)) \\ h_2(k+1) &= h_2(k) + t_s(Flow_{low}/A) \qquad (25) \\ h_3(k+1) &= h_3(k) + t_s((h_1(k) - h_3(k))/\tau_2(k) - h_3(k)/\tau_3(k)) \end{aligned}$$

where sampling time $t_s = 2s$, A is cross section area of tanks, and $\tau_1(k), \tau_2(k)$, and $\tau_3(k)$ are state-dependent time constants at kth instant. For the limit of space, refer to [10] for more details on the parameters.

A series of 200-second-simulation experiments have been conducted using the model equations like (25) in which the exhaustive search and sDFS are used for horizon of 3, 4, 5 and 6. The control goal is to reach and maintain the three fluid heights at {0.3m, 0.25m, 0.15m}. The initial point is at {0.2m, 0.18m, 0.1m}. The control goals are all realized. Table 3 shows the results, where * denotes a huge number of 25,646,166 nodes that need to be visited using exhaustive search at a single time step for horizon of 6. In contrast to this huge and impractical amount, the sorted depth search is able to provide significantly smaller numbers which makes the controller computations still doable. The similar situations can be found for the cases of horizon of 4 and 5. As is shown, the sDFS algorithm turns a typically heavy computation into a much more manageable task.

TABLE 3: EXHAUSTIVE SEARCH AND SORTED DEPTH FIRST SEARCH FOR EXPERIMENT 2

Horizon	Point A	Point B	Point C
3	5,219	2,516/595	16%
4	88,740	13,396/1,377	3.1%
5	1,508,597	44,608/3,009	0.6%
6	*	124,066/7,310	-

V. CONCLUSIONS

In this paper, a finite-input receding horizon controller is presented to deal with a kind of switching control systems where only a finite number of discrete-valued controls are available. The controller design is motivated by the need of motion control of a pneumatic hopping robot using on/off solenoid valves. The pneumatic motion control problem is studied and recast as an energy regulation problem. A desired total conservative energy is computed and serves as an objective in the control scheme. Simulation results demonstrate the effectiveness of the proposed (FIRHC) controller. Stability analysis is discussed. The terminal equality constraint is added to ensure the stability, but it leads to severe restriction of the operation region of the system. More effective stability condition needs to be developed.

In addition to the controller design, this paper develops an efficient approach, sorted depth first search (sDFS), to seek solutions of the optimization problem in the proposed controller. The algorithm takes advantage of the fact of cost always being positive and expands first the nodes with lower cost to avoid unnecessary computation. It has been effectively applied for the motion control system, and for the further demonstration it is used for a three tank system that has a relatively large amount of operation modes. The simulation results show that compared to exhaustive search, the previously generally used method in the same kind of problem, sDFS achieves much better efficiency in both cases.

REFERENCES

[1] M. Raibert, *Legged Robots that Balance*, MIT Press, Cambridge, MA, 1986.

[2] M. Binnard, *Design of a Small Pneumatic Walking Robot*, master's thesis, MIT, Cambridge, MA, 1995.

[3] N. M'sirdi, N. Manamani, and N. Nadjar-Gauthier, "Methodology based on CLC for Control of Fast Legged Robots," *Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robotics and Systems,* Victoria, B.C., Canada, pp. 71-76, 1998.

[4] Y. Zhu, and E. Barth, "An Energetic control methodology for pneumatic hopping robot," submitted to IEEE/ASME Transactions on Mechatronics, 2006.

[5] E. Camacho and C. Bordons. Model Predictive Control, Springer, London, 1999.

[6] J. Maciejowski, Predictive Control with Constraints, Prentice Hall, London, 2002.

[7] S. J. Qin and T. A. Badgwell. "An overview of industrial model predictive control technology," in J. C. Kantor, C. E. Garcia, & B. Carnahan (Eds.), Chemical process control—V, Fifth international conference on chemical process control CACHE and AICHE, pp. 232–256, 1997.

[8] S. J. Qin and T. A. Badgwell. "An overview of nonlinear model predictive control applications." in F. Algower, & A. Zheng (Eds.), Nonlinear model predictive control workshop—assessment and future directions, Ascona, Switzerland, 2000.

[9] T. Tsang, and D. Clarke, "Generalized Predictive Control With Input Constraints," IEEE Proceedings—Control Theory and Applications, 135(6), pp. 451–460, 1988.

[10] J. Wu, S. Abdelwahed, and G. Biswas, "A Hybrid Control System Design and Implementation for a Three Tank Test-bed", in *Proc. IEEE Conference on Control Applications*, pp. 645 – 650, Toronto, Canada, September 2005.

[11] S. Abdelwahed, J. Wu, G. Biswas, and E. Manders, "Online Fault Adaptive Control for Efficient Resource Management in Advanced Life Support System," *Habitation – International Journal for Human Support Research*, 10(2): 105 – 115, 2005.

[12] B. Shields, E. Barth, and M. Goldfarb, "Predictive Control for Time Delayed Switching Control Systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, 999-1004, Dec. 2006.

[13] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, Second Edition. pp.540–549, MIT Press and McGraw-Hill, 2001.

[14] D. Ben-Dov, and S. Salcudean, ''A Force-Controlled Pneumatic Actuator,'' *IEEE Trans. Robotic Automation*, vol. 11, No. 6, pp. 906–911, 1995.

Jian Wu was born in Harbin, China. He received a Master of Engineering degree in control theory and applications from Harbin Institute of Technology, Harbin, China in 1997, and a Master of Science degree in Electrical Engineering from Vanderbilt University, Nashville, TN in 2007.

He is currently a controls analyst with Omron Oilfield and Marine in Houston, TX. His major research interests involve AC/DC drive systems modeling, control and simulation, digital control techniques of nonlinear systems, and model predictive control theory and applications.

Sherif Abdelwahed is an Assistant Professor with Electrical and Computer Engineering Department at Mississippi State University. He received his Ph.D. degree in Electrical and Computer Engineering from the University of Toronto, Canada, in 2002. From 2000 to 2001, he was a research scientist at Rockwell Scientific Company. From 2002 to 2007 he worked as a research assistant professor with the Institute for Software Integrated Systems at Vanderbilt University. His main research interests include model-based design and analysis of self-managing computation systems, modeling and analysis of distributed real-time systems, automated verification, fault diagnosis techniques, and model-integrated computing. He has published over 70 publications. He is a senior member of the IEEE and member of Sigma Xi.