

Design, development and control of a hopping machine – an exercise in biomechatronics

Kuldip Naik^a, Mehran Mehrandehz^{b*} and John Barden^c

^a*iQmetrix, Regina, SK, Canada*; ^b*Faculty of Engineering, University of Regina, Canada*; ^c*Faculty of Kinesiology and Health Studies, University of Regina, Canada*

(Received 14 November 2008; final version received 2 September 2009)

Hopping is a complicated dynamic behaviour in the animal kingdom. Development of a hopping machine that can mimic the biomechanics of jumping in *Homo sapiens* is envisioned. In this context, the design, development and control of a cost-effective, pneumatically actuated, one-legged hopping machine were initiated at the University of Regina in 2005. The pneumatic actuator has a simple design that employs an off-the-shelf on/off control valve which regulates the air pressure supplied to the hopper's body using a pulse width modulated (PWM) signal. The objective is to maintain a constant jumping height in the hopper after going through a finite number of hopping cycles. The mechanistic model of the system was investigated in full detail. This model facilitates: (1) the design of the actuating system, and (2) the synthesis and verification of different control strategies in a simulation environment prior to implementation in the real world. The movement of the hopper is supported by a vertical slide; therefore, the hopper can only jump in place. However, the proposed control strategy and the propulsion unit can be further utilised for stable hopping in a 3-D environment. A model-free Neuro-PD controller was then designed, trained and implemented on a real system. Simulation and experimentation showed promising results. This system can be used as an educational tool for teaching real-time control of hybrid and non-linear systems. It can be also used as a biomechatronics test bed to simulate the effect of different timings in firing action potentials in jump-causing leg muscles on achieving a desired jumping height in the animal kingdom.

Keywords: hopping robots; biomechatronics; real-time control; artificial neural networks; mathematical modelling of dynamical systems; hybrid control systems

1. Introduction

Research into legged robots began about two decades ago. However, it has been vastly revisited in the new millennium due to the recent interest in the development of rovers suitable for exploration of rough extraterrestrial terrains. Research in legged robots also has the potential to lead to development of machines which will be useful in fields of transportation, forestry, agriculture, fire fighting, defense (carrying weapons, de-mining), urban policing, assistive devices for walking, entertainment, service robotics and biomechanics. The proposed platform can serve as a test bed to examine different hopping strategies adopted in the animal kingdom. For instance, the effect of the different timings in firing the action potentials in leg muscles for high-jumping animals and for humans can be compared and evaluated within a lab environment using this platform by simply changing the timing of the pressurisation and depressurisation of the pneumatic valve.

Hopping machines were extensively studied by Raibert in the 1980s. Most of these robots used a telescopic leg with an internal air spring as compliance in series with a hydraulic thrust actuator (Raibert 1984). Papantoniou (1991) also proposed an electromechanical design of a one-legged

planar hopping robot. Their robotic leg was constructed of a four-bar linkage with a tension spring. Buehler (2002) employed electric actuators for a one-legged planar hopper which relied on passive dynamics of the system and radially compliant leg designs. Sato and Buehler (2004) presented control of a 2-DOF hopping robot with a single actuator based on the spring loaded inverted pendulum (SLIP) model with the help of a PD controller. The leg angle was controlled with the help of torque applied by a motor connected at the hip joint. Lebaudy, Prosser and Kam (1993) designed a vertically constrained, electrically actuated hopping machine with a telescopic leg. They presented three different control algorithms in which the near-inverse controller with integral error feedback demonstrated the best tracking performance in the presence of abrupt mass changes. Sznair and Damborg (1989) used an adaptive control algorithm based on online numerical minimisation of the performance criterion to control both vertical and horizontal motion in a 2-D hopping robot. Prosser and Kam (1992) suggested a control algorithm to regulate the jumping height of an electrically actuated hopper with a linear elastic element (spring) in the leg, based on offline synthesis of a near-inverse model for the plant achieved by numerical

*Corresponding author. Email: Mehran.Mehrandehz@uregina.ca

simulation of the system dynamics. Mehrandezh et al. (1995) proposed a method to control the jumping height of an electrically actuated one-legged hopping robot. The minimum number of hops a robot would need to execute before reaching a desired jumping height was estimated. To eliminate sensitivity due to perturbation of the system parameters, a modified PI controller was implemented which regulated the jumping height of the robot. For further information, a detailed review of research pertaining to hopping robots has been presented by Sayyad et al. (2007).

Like most other research (Helferty et al. 1989; Sznair and Damborg 1989; Mehrandezh et al. 1995), our prototype's motion was limited to a single plane. However, the mechanics of this system were simpler than the mechanics of its counterparts. The pneumatic actuator is controlled in real-time by an on/off valve via a PWM signal. An infrared (IR) and a *linear optical encoder* were used to measure jumping height and the stroke of the single-acting cylinder used in the hopper, respectively. An important aspect of the hopping robot is that the natural dynamics of the plant are responsible for producing the hopping motion. Even with no actuation, the prototype can still bounce several times when dropped on the ground before settling down. This one-legged hopping robot embodies a minimalist design approach that emphasises the design of mechanical oscillators with motions that will accomplish a task, which leads to simplicity and energy efficiency. The slim geometry of one-legged robots makes them useful to move in narrow passages and/or confined spaces. Our hopper is 50 cm tall when at rest and 65 cm tall when its pneumatic cylinder (i.e. body) is fully stretched.

Figure 1 shows the pneumatically actuated one-legged hopping robot built at the University of Regina. The continuous pressurisation (i.e. the phase through which the air is allowed to flow from the air compressor to the air cylinder used as the body of the hopper) and depressurisation (i.e. the phase through which air from the air cylinder is allowed to exhaust to the atmosphere) phases enable the hopping robot to jump continuously. The reference variable is the maximum jumping height to be achieved and maintained

by the robot after going through a finite number of hops. The control signal (e.g. the actuator's stroke at which the pressurisation starts) is computed at the start of every hop and is then utilised in the next hopping cycle.

The salient contributions of this paper are: (1) design of a simple one-legged hopping machine, (2) development of a comprehensive mechanistic model of the hopper that can be utilised for optimising the mechanical design, controller and the sensor pack, (3) building a cost-effective hopping machine using off-the-shelf on-off pneumatic valves, (4) building an educational platform that can be employed to study the effect of different jumping actions adopted in the animal kingdom in a lab environment and (5) devising a control strategy that can bring the robot to its desired jumping height in the shortest time. This is especially important in applications where maintaining balance quickly in near-fall scenarios is desirable. A practical example could be a climb up a staircase with narrow steps.

This article describes the hopping robot as follows: The problem statement is given in Section 2; Section 3 describes the mechanistic model of the hopping robot used in the simulation; open- and closed-loop control of the robot using its mechanistic model through computer simulations are explained in Section 4; the experimental setup and the control of the real prototype are investigated in Sections 5 and 6, respectively; the experimental results are given in Section 7 and conclusions and future work are given in Section 8.

2. Problem statement

The objective of this research was to develop a hardware-in-the-loop (HIL) real-time controller to regulate the jumping height of a one-legged hopping robot whose motion was supported by a vertical stand. A second objective was to have the controller achieve the desired jumping height in the shortest possible time and to maintain this height by adjusting the air pressure inside the pneumatic cylinder, which acts as the body of the hopping robot, at each hopping cycle.

The control of the hopping robot represents a nonlinear and hybrid problem.¹ The presence of friction between the body and the leg, delay (or deadband) in the pneumatic actuation systems and the dynamic impact between the body and the leg and the leg and the ground, make it a challenging control problem. The goal was to demonstrate the capability of a real-time controller to regulate the jumping height of the hopper based on user-defined set points in the presence of the aforementioned disturbances and uncertainties. Among all available forms of actuators, pneumatic actuators were selected because they offer a higher power to mass ratio than their counterparts at a lower cost.

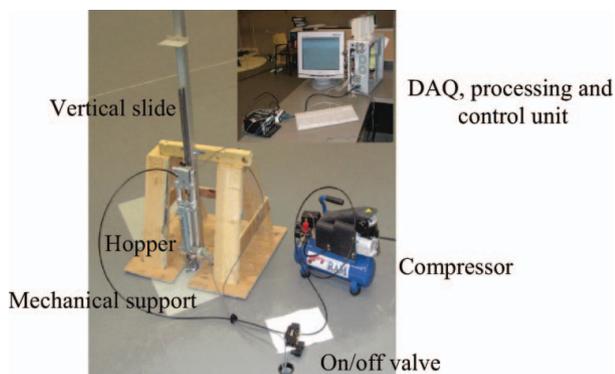


Figure 1. The experimental pneumatically actuated hopping robot.

¹In general, hybrid structures can be found in control algorithms as well as in the description of the physical behaviour of processes. A hopping machine has a hybrid structure in both the control and process domains.

3. Mechanistic model of the robot

A comprehensive mechanistic model of our hopper was developed. This model was then used to (1) size the pneumatic actuator, (2) to synthesise different control strategies in a simulation environment and (3) to do a qualitative sensitivity analysis offline in a simulation environment to select the most appropriate on-board sensors for feedback purposes, prior to implementation in the real world. The robot had the following components: a single-acting pneumatic cylinder (body), a piston that can move inside the cylinder (leg), an on-off control valve to regulate the pressure inside the cylinder and a pressure supply (i.e. an air compressor) which was connected to the pneumatic cylinder (body) of the hopper via the on/off control valve operated by a pulse width modulated (PWM) signal. The body of the hopper acted as an actuator as well as a passive spring (when the valve was closed) in the system. It should be noted that by controlling the duty cycle of the PWM signal supplied to the on/off valve, the pressure inside the body of the hopper could be controlled. An on/off valve was used, as opposed to a proportional solenoid valve, because of its simple construction and cost effectiveness without compromising the overall performance of the system. Figure 2 shows all parameters

used in the development of the mechanistic model for the hopping robot.

The kinematics equations relating the stroke in the single-acting pneumatic cylinder, x , to the movement of the body and leg COGs, namely y_1 , and y_2 , are:

$$x = C + (y_2 - y_1) \tag{1a}$$

$$y_1 = y_0 + \frac{L_L}{2}. \tag{1b}$$

In Equation (1a), C is a constant. In the case that the centroid of the hopper's leg and body are located at their medians, then one can write: $C = (L_b - L_L)/2$.

The dynamics equation governing the motion of the hopper's leg is as follows:

$$-M_1g - F + F_G + F_{M2} - F_{M1} - F_f - \beta \dot{x} = M_1 \ddot{y}_1, \tag{2}$$

where F denotes the force acting on the top area of the leg due to the resultant differential pressure (i.e. the difference between the air pressure in the body, P , and the atmospheric pressure, P_a , (i.e. $F = A (P - P_a)$) and $\beta \dot{x}$ and F_f are the resistive forces generated by the viscous and

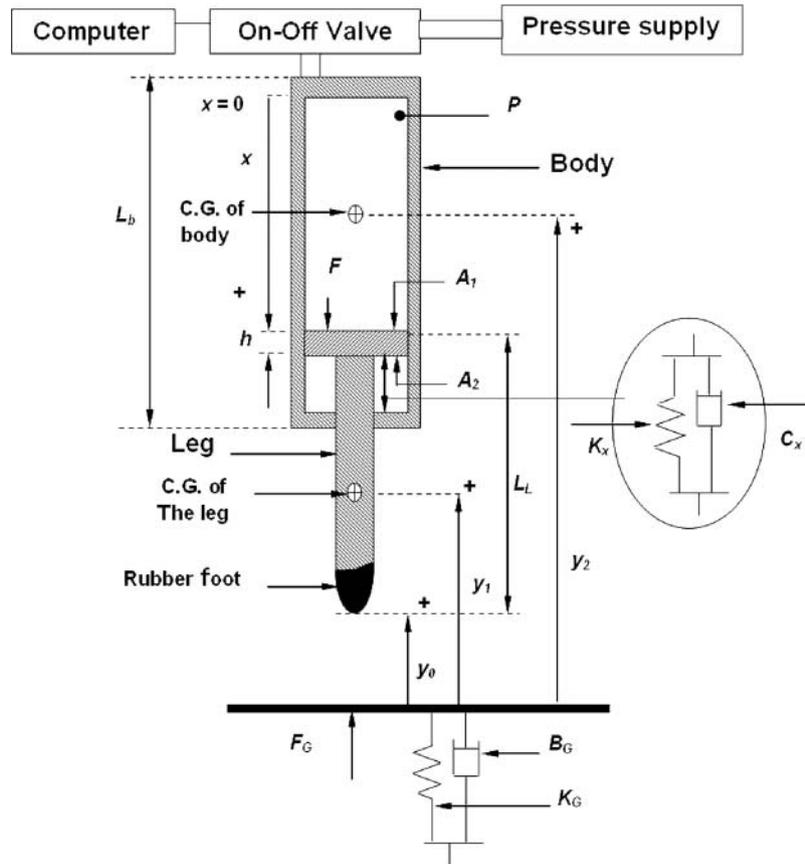


Figure 2. Internal structure of the hopping robot.

coulomb frictions, respectively. F_{M1} and F_{M2} are the forces generated due to the mechanical impacts between the upper/lower end of the body and the leg, respectively. M_1 and g in Equation (2) denote the leg's mass and the gravitational acceleration, respectively. F_G is the force exerted on the leg by the ground. This force was modelled as a spring/damper system as follows:

$$F_G = \begin{cases} -K_G y_0 - B_G \dot{y}_0 & \text{if } y_0 < 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The friction force, F_f is calculated as:

$$F_f = \begin{cases} F_{sf} & \text{if } \dot{x} = 0 \\ F_{df} \text{sign}(\dot{x}) & \text{if } \dot{x} \neq 0 \end{cases} \quad (4)$$

where F_{sf} and F_{df} in Equation (4) denote the static and dynamic dry frictions, respectively. The impact forces F_{M1} and F_{M2} in Equation (2) are calculated as:

$$F_{M1} = \begin{cases} -K_x x - C_x \dot{x} & (x \leq 0) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

$$F_{M2} = \begin{cases} K_x (x - L_b + h) + C_x \dot{x} & (x > L_b - h) \\ 0 & \text{otherwise} \end{cases}$$

The rate of change of pressure P through the pressurisation and depressurisation phases were modelled as follows (Richer and Hurmuzulu 2000):

$$\dot{P}_{in} = \frac{P}{V_c} [K_{in} A_v \dot{m}_p - \dot{x} \alpha A]$$

$$\dot{P}_{out} = -\frac{P}{V_c} [K_{out} A_v \dot{m}_{dp} + \dot{x} \alpha A]. \quad (6)$$

In Equation (6), V_c denotes the active volume (or the expanding volume) of the hopper's cylinder. It can be calculated as follows:

$$V_c = Ax. \quad (7)$$

K_{in} and K_{out} in Equation (6) represent the combined effect of the supply pressure, P_s , the heat transfer coefficient for air, α , the valve discharge coefficient, φ , and the absolute air temperature, T . They can be calculated as follows:

$$K_{in} = C_f R \alpha_{in} \varphi_{in} P_s \sqrt{T} \quad K_{out} = C_f R \alpha_{ex} \varphi_{ex} \sqrt{T}. \quad (8)$$

In Equation (8), $\alpha, \alpha_{in}, \alpha_{ex}$ denote the heat transfer coefficients for air at three stages, namely the air residing inside the cylinder, the inflow and the air exhausted to the atmosphere, respectively. Given that the pressurisation and depressurisation processes are carried out very quickly, it can be assumed that these coefficients remain constant throughout the entire process. The in/out flow attenuation

components, namely φ_{in} and φ_{ex} given in Equation (8), can be also calculated as:

$$\varphi_{in} = \exp\left(\frac{-R_t R T L_t}{2 P_s C}\right) \quad \varphi_{ex} = \exp\left(\frac{-R_t R T L_t}{2 P C}\right), \quad (9)$$

where R_t is the air hose's resistance to the flow of air (which varies according to the nature of the flow), L_t is the hose length, and C denotes the velocity of sound. Correspondingly, depending on the flow regime, the hose's resistance to the airflow can be calculated as follows:

$$R_t = \begin{cases} 32\mu/D^2 & \text{For laminar flow} \\ \frac{0.158 \mu R_e \frac{3}{4}}{D^2} & \text{For turbulent flow} \end{cases}, \quad (10)$$

In Equation (10), represents the dynamic viscosity of the air, D denotes the internal diameter of the air hose and R_e is the Reynolds number calculated for the inflow. The mass flow rates given in Equation (6), namely \dot{m}_p and \dot{m}_{dp} , represent the mass flow rate of air through the valve during the pressurisation and depressurisation phases, respectively, where:

$$\dot{m}_p = \begin{cases} C_1 & \text{if } P/P_s \leq P_{cr} \\ C_2 (P/P_s)^{\frac{1}{k}} \sqrt{1 - (P/P_s)^{\frac{k-1}{k}}} & \text{if } P/P_s > P_{cr} \end{cases}, \quad (11)$$

$$\dot{m}_{dp} = \begin{cases} C_1 & \text{if } P_a/P \leq P_{cr} \\ C_2 (P_a/P)^{\frac{1}{k}} \sqrt{1 - (P_a/P)^{\frac{k-1}{k}}} & \text{if } P_a/P > P_{cr} \end{cases}. \quad (12)$$

In Equations (11) and (12), k represents the specific heat ratio for air; P_{cr} , C_1 and C_2 are constants calculated as follows:

$$P_{cr} = (2/(k-1))^{(k-1)/k},$$

$$C_1 = \sqrt{(k/R)(2/(k+1))^{(k+1)/(k-1)}},$$

$$C_2 = \sqrt{2k/(R(k+1))} \quad (13)$$

Correspondingly, the dynamics equation for the body of the hopper is as follows:

$$-M_2 g + F + F_{M1} - F_{M2} - F_f - \beta \dot{x} = M_2 \ddot{y}_2. \quad (14)$$

The physical impact between the ground surface and the foot (bottom of the leg) of the hopping robot was modelled as a spring with stiffness K_G and a damper with damping coefficient B_G . Also, the physical impact between the leg and the body of the hopper was modelled as a spring with stiffness K_X and a damper with a damping coefficient B_X .

The mechanistic model of the robot enabled us to design the actuation system by analysing the effect of different

control strategies/variables on maintaining a constant jumping height. It was also used to size the pneumatic actuator and the air supply unit (i.e. the pressure range required) in a simulated environment. The procedure for model-based design of the controller is described in Section 4 with additional details.

4. Numerical simulation

Numerical simulation of the mechanistic model enabled us to conduct a feasibility study on the possible control strategies required to regulate the jumping height of the hopping robot. It also enabled us to analyse the effects of various factors such as friction, actuation time delay, dynamic impact between mechanical parts and mass flow attenuation due to resistance of the air hose. The components required for the functioning of the hopping robot were either purchased or designed and built in our lab. A commercial air pogo stick was utilised as the basic framework of the system. The values associated with the physical parameters used to model the hopping robot were chosen on the basis of the manufacturer's specifications. Some parameters such as the mass of the leg, M_1 , the mass of the body, M_2 , and the stroke of the pneumatic cylinder, x , were directly derived from the actual measurements of the air pogo stick. The valve parameters such as the valve coefficient, time delay in the actuation, etc. were calculated based on the specifications of the Norgren's Nugget 200, 1/4" solenoid, pilot actuated, 3 port, 2 way on-off control valve (Norgren 2009). Other parameters such as those associated with the mathematical model of the friction and the dynamic impact between the robot and the ground were estimated through open-loop tests on the real hopper. Table 1 summarises the physical parameters and their corresponding numerical values set in the simulation. Further details on system identification and control design are described in Sections 4.1 and 4.2, respectively.

4.1. System identification through open-loop testing

The unspecified parameters of the mechanistic model, such as friction, dissipation of energy due to mechanical impacts and air compressibility were identified through open loop tests. In these tests, the first step was to keep the valve closed while hopping. Therefore, the air trapped inside the cylinder acted as a passive spring. The behaviour of the system was recorded in real-time and compared with the simulated model offline. The hard-to-measure parameters mentioned above were then identified (i.e. estimated) through a trial-and-error procedure by matching the real and the simulation results. In these open-loop tests, the hopper was released from a certain height while keeping the control valve closed all the time (i.e. no pressurisation and depressurisation). The rate of change of pressure inside the hopper's body, under

Table 1. Physical parameters and their numerical values used in the simulation.

Parameter	Symbol	Value and unit
Body mass of the hopping robot	M_2	1.3 kg
Mass of the leg of the hopper	M_1	1 kg
Body length of the hopping robot	L_b	0.8 m
Circular area of the body of the hopper	A	0.1963 m ²
Upper area of the leg	A_1	0.001257 m ²
Lower area of the leg	A_2	0.000628 m ²
Length of the leg of the hopper	L_L	1 m
Supply pressure	P_s	137.895 kpa
Atmospheric pressure	P_a	101.325 kpa
On-off valve coefficient (discharge/charge constant of the valve)	C_{va}	15 N ^{1/2} /m-s
Compressibility of the air	K_c	2 N/m ²
Non-dimensional, geometry dependent, discharge coefficient	C_f	0.82
Spring coefficient of the spring used to model the impact between the leg and the body	K_x	500 N/m
Damping coefficient of the damper used to model the impact between the leg and the body	C_x	125 N-s/m
Spring coefficient of the spring used to model the impact between the rubber foot and the ground	K_G	15000 N/m
Damping coefficient of the damper used to model the impact between the rubber foot and the ground	B_G	125 N-s/m
On-off valve's port area	A_v	0.001269 m ²
Length of the hose supplying air to the body of the hopping robot	L_t	3.0 m
Static friction force	F_{sf}	21 N
Dynamic friction force	F_{df}	10 N
Dynamic viscosity	μ	2.066 N-s/m ²
Air temperature in the body of the robot	T	293 K
Ideal gas constant	R	8.31 L-atm/K-mole
Diameter of the pneumatic hose supplying air to the body from the pressure supply	D	0.0032 m
Heat transfer coefficient of air during the pressurisation phase	α_{in}	1.4
Heat transfer coefficient for the depressurisation phase	α_{ex}	0.9
General heat transfer coefficient of the air	α	1.2
Specific heat ratio	k	1.4
Velocity of sound	C	340 m/s

the assumption that the air compression/expansion remain adiabatic, can be calculated as:

$$\dot{P} = \frac{-nRT\dot{V}}{V^2}, \quad (15)$$

where R is the universal gas constant, n denotes the number of gas molecules in moles and T denotes the air absolute temperature (assumed to remain constant throughout the hopping). Equation (15) in conjunction with Equations (1)–(5) were then numerically simulated and the result was compared with the real system's response in order to estimate the dry and viscous friction coefficients. For further detail on the system identification process please refer to Naik (2006).

4.2. Design and evaluation of a PID controller

A number of control strategies were synthesised while pressurising/depressurising the hopper at a certain rate to: (1) size the actuator and the air supply unit and (2) devise a robust control strategy that could be implemented in the real system. A simple PID controller was deemed adequate. Therefore, a PID controller was designed, tested and tuned in a simulation environment. The control variable was chosen as the *duty cycle* (or dc for brevity) of the PWM signal provided to the on/off control valve. The calculated duty cycle at each hopping cycle was cropped to its limits (i.e. dc_{\min} and dc_{\max}) as follows:

$$dc = \begin{cases} dc_{\max} & \text{if } dc \geq dc_{\max} \\ dc & \text{if } dc_{\min} < dc < dc_{\max} \\ dc_{\min} & \text{if } dc \leq dc_{\min} \end{cases} . \quad (16)$$

The following updating law for the duty cycle applied to the control valve based on a PID controller was

adopted:

$$dc_{i+1} = dc_i + K_P(H_d - H_i) + K_D(H_i - H_{i-1}) + K_I \sum_{j=1}^i (H_d - H_j), \quad (17)$$

where K_P , K_D and K_I denote the Proportional, Derivative and Integral gains, respectively. The well-known Ziegler–Nichols step response method was employed to tune the PID gains (Ogata 2002). In Equation (17), H_i denotes the jumping height at the i th hopping cycle, dc_i denotes the PWM's duty cycle at the i th hopping cycle and H_d denotes the desired jumping height. An *anti-windup* strategy was also adopted to avoid actuator saturation due to the integral term. For further details one should refer to Naik (2006). Figure 3 shows the simulation results when employing this PID controller. It is noteworthy that despite the fact that the hopping motion has a cyclic nature, but the maximum jumping height, as can be seen in Figure 3, increases exponentially to the desired jumping height, which is the characteristic of an under damped second-order system. Furthermore, the small non-zero steady-state error in jumping height observed in Figure 3 is due to the physical impact between robot parts and that with the ground.

It can be observed that the time taken by the controller to achieve the desired jumping height (that is, foot elevation) of 0.2 meters was approximately 5 seconds and the desired jumping height was achieved within 5 hops with a near-zero steady-state error. The number of hops required before reaching the steady state can be reduced by incorporating an Inverse-Dynamic (ID) based approach within

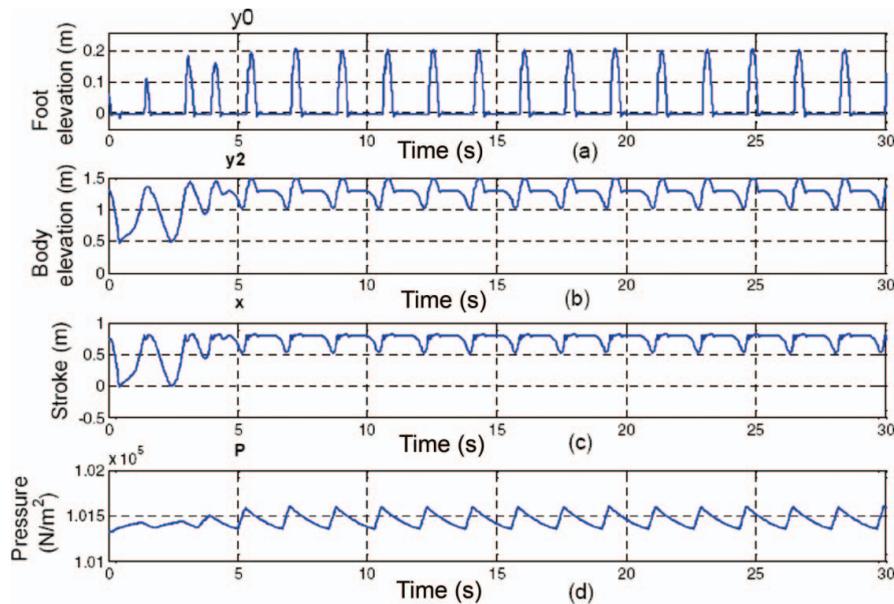


Figure 3. Simulation results of the PID controller.

the control law. An *Artificial Neural-Network* (ANN) was utilised to synthesise the inverse dynamic approach subjected to unmodelled errors in the system. The output of the ANN-based ID approach was then incorporated into the control strategy. The details of this control strategy will be described in Section 6.

Although the suggested PID controller was capable of controlling the jumping height of the hopping robot, some problems associated with the timing of the control strategy were discovered during the computer simulations. For instance, it was realised that pressurising the cylinder at the touchdown phase (the phase at which the leg hits the ground) and depressurising it at the lift-off phase (the phase at which the hopper loses contact with the ground) could induce unwanted vibrations with fast dynamics (i.e. resonance). The cause of this resonance was the mechanical impact between the leg and the ground at the touchdown and also between the leg and body at the lift-off phase. Figure 4 shows this phenomenon. From this it was concluded that the touchdown and lift-off phases might not be the most appropriate times to pressurise/depressurise the system. Based on this fact, another control strategy was adopted in the real system which will be described in Section 6.

5. Experimental setup

Our hopping robot was able to jump in place along a vertical support (i.e. a linear slide). The body of the hopping robot was fitted onto a mounting plate to provide vertical motion on the linear slide. The air was supplied to the hopper from a floor compressor (0-120 psi) via a 3/2-port on/off control valve adopted for regulating the pressure supplied to the hopper with the help of a 1/4" lightweight air hose. A 6036-PCI card from National Instruments was used

as the DAQ system (National Instruments 2008). A linear quadratic optical encoder was used to measure the stroke, x , and an IR sensor was used to measure the body's elevation, y_2 . A low-pass filter was designed to filter the noise in the IR readings.

The hopping robot is considered to be a discrete control system, where one cycle of motion is regarded as one sampling interval of the controller. However, the state of the system was updated at a much faster rate. The Real-Time Windows Target (RTWT) and Real-time Workshop toolboxes along with Simulink from Mathworks were used for the real-time control of the system. A C-code for real-time operation was generated directly from Simulink via the Open Watcom C/C++ compiler.

6. Control of the hopping robot

A Neuro-PD controller was developed for the real system which outperformed a PD and/or PID controller in terms of the response time. The *neural network* was specifically utilised to estimate the inverse dynamics of the system. Furthermore, to avoid resonance in the hopper's motion, the timing for the pressurisation was carried out based on the stroke, x , in the real system. This way unwanted vibration in the optical encoder that could lead to faulty results was avoided. The overall structure of the Neuro-PD controller with a modified timing constitution is described below.

The error input to the controller was calculated as:

$$E_i = H_d - H_i, \quad (18)$$

where E_i denotes the error between the desired maximum jumping height, H_d , and the maximum jumping height (i.e. peak height) at the i th hop, H_i . The only control variable was the stroke, x , at which the pressurisation occurred. In

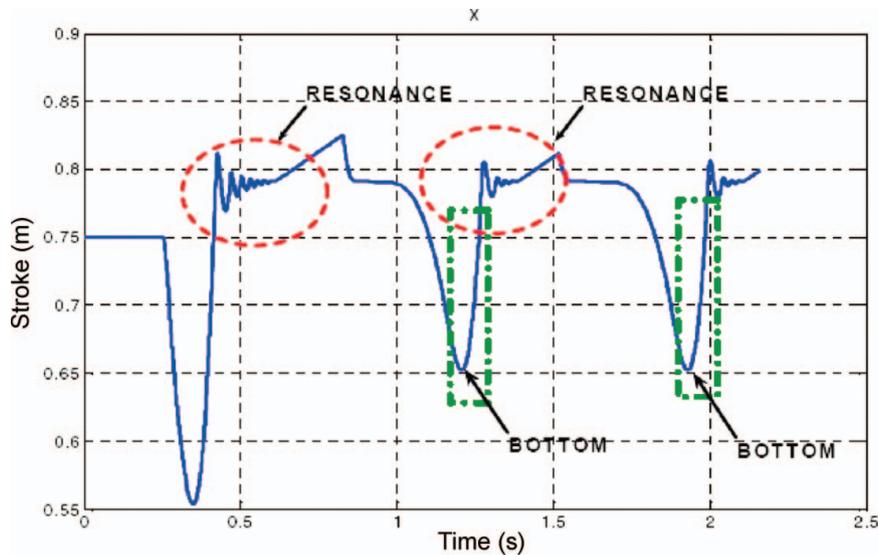


Figure 4. Time variation of the body's stroke. Mechanical impact causes vibration.

this control strategy, it was assumed that the duty cycle of the PWM signal provided to the control valve remained constant during the pressurisation process. A number of tests were conducted with different air pressures supplied by the compressor and it was verified that the control variable, x , at which pressurisation starts had a significant impact on the jumping height achieved by the robot. A stroke-based control strategy was then developed based on the following steps. First, the control variable x was cropped based on its hard limits (i.e. x_{\min} and x_{\max}) as follows:

$$x = \begin{cases} x_{\max} & \text{if } x > x_{\max} \\ x & \text{if } x_{\min} < x < x_{\max} \\ x_{\min} & \text{if } x < x_{\min} \end{cases} \quad (19)$$

Second, a PD controller with an offset term was designed as:

$$x_{i+1} = x_i - K_P(H_d - H_i) + K_D(H_i - H_{i-1}), \quad (20)$$

where, x_i denotes the stroke value at which the system was pressurised in the i th hop. Controller gains, namely K_P and K_D , were tuned with the help of the Ziegler–Nichols step response method (Ogata 2002). Figure 5 shows the experimental results when using a stand-alone P and PD controller. In Figure 5 the desired jumping height of the hopper's body (i.e. body elevation) was set at 82 centimetres. It can be observed from Figure 5 that the hopper was able to achieve sustainable hopping while maintaining the

user-defined maximum jumping height for the time span of 20 seconds. Lesser overshoots can be observed in Figure 5 when the PD controller was used. The number of jumps required to settle to the desired jumping height, within a certain threshold, when employing a PD controller was 5 where that for a P controller was 9 with sporadic violations in maintaining the jumping height close to the desired at the 13th, 16th, 25th and 26th jumps as well.

Through simulation, it was conjectured that faster responses would be possible if an inverse-dynamic based approach was incorporated into the controller. Subsequently, an ANN was designed to model the inverse dynamics of the system. Further details of this approach are provided in Section 6.1.

6.1. Neural-network-based inverse dynamics

In the present context, inverse dynamics refers to the process of predicting a control strategy (in the form of the required actuation effort) that would bring the system from an initial state to a desired terminal state. As such, the inverse dynamics approach can be viewed as a predictive control strategy. The amalgamation of a predictive and a reactive feedback control strategy (e.g. a PD controller) should result in performance that exceeds that of the individual controllers. Although the predictive control strategy could have been synthesised using the mechanistic model presented in Section 3, due to unmodelled dynamics pertinent to friction, physical impacts and air compressibility,

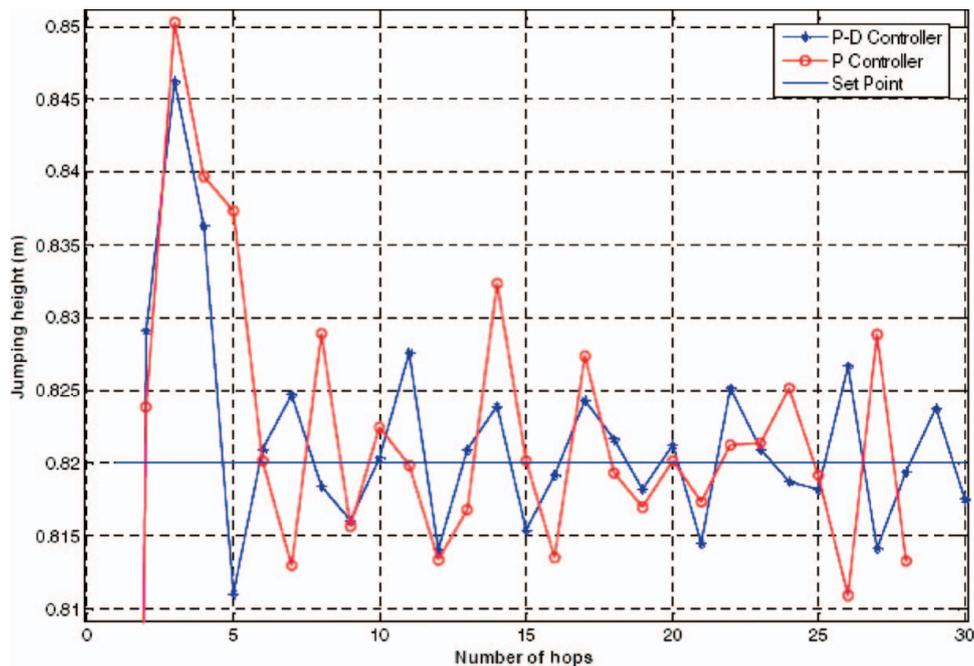


Figure 5. Comparison of the P and PD controller applied to the real hopping robot. Note that the jumping height represents the elevation of the body's COG.

we decided to utilise a model-free neural-network-based approach so that the inverse dynamics of the system could be modelled with higher precision using the real prototype. ANNs have a remarkable ability to learn and derive meaning from complicated or imprecise data which can be used to extract information and detect trends that are too complex to be noticed by either humans or other computer techniques. In general, artificial neural networks are incorporated into a control strategy as follows: A neural network model of a nonlinear system is derived and trained (online and/or offline) to predict future plant performance. The controller then calculates the control input on fly that will optimise plant performance over a specified future time horizon (in our case this horizon is defined by the number of jumps required to reach the desired jumping height). Figures 6a and 6b show the block diagram of the proposed ANN-based control strategy.

Neural networks have been applied to the identification and control of linear and non-linear dynamic systems. For example, Werbos (1991) specified a direct inverse control method, in which the ANN learns the inverse dynamics of the system, so that it can command the system to follow a user specified trajectory. Helferty et al. (1989) demonstrated a neuron-like learning/control system, applied to a 1-DOF, one-legged electrically actuated hopping robot with

a springy leg whose desired motion was characterised by periodicity in the state space variables. The difference between our research and the work done by Helferty et al. is that they applied the neuromorphic controller to the mathematical model of an electrically actuated hopping robot, whereas we applied the ANN-based controller to the real hopper. In our control strategy, a 3-state vector, s , is defined as:

$$s = [x \dot{x} y_2]^T, \quad (21)$$

where the optimal value of the stroke, x , to bring the robot to its desired jumping height is estimated through a trained neural network based on the state vector s given in Equation (21). It is noteworthy that the sequential effect of the pressurisation and depressurisation phases in training the ANN-based inverse dynamic approach was kept at minimum by tuning the timing of these phases. In fact, the timing of pressurisation and depressurisation were tuned offline in a way that it would guarantee no change in internal states of the hopping machine going from one jumping cycle to another. In other words, it was assured that whenever the robot reaches its peak during its flying phase, the inside pressure has been already reduced to the atmospheric pressure and the pneumatic cylinder has been retracted to its full length. Therefore, there will be no residual effect

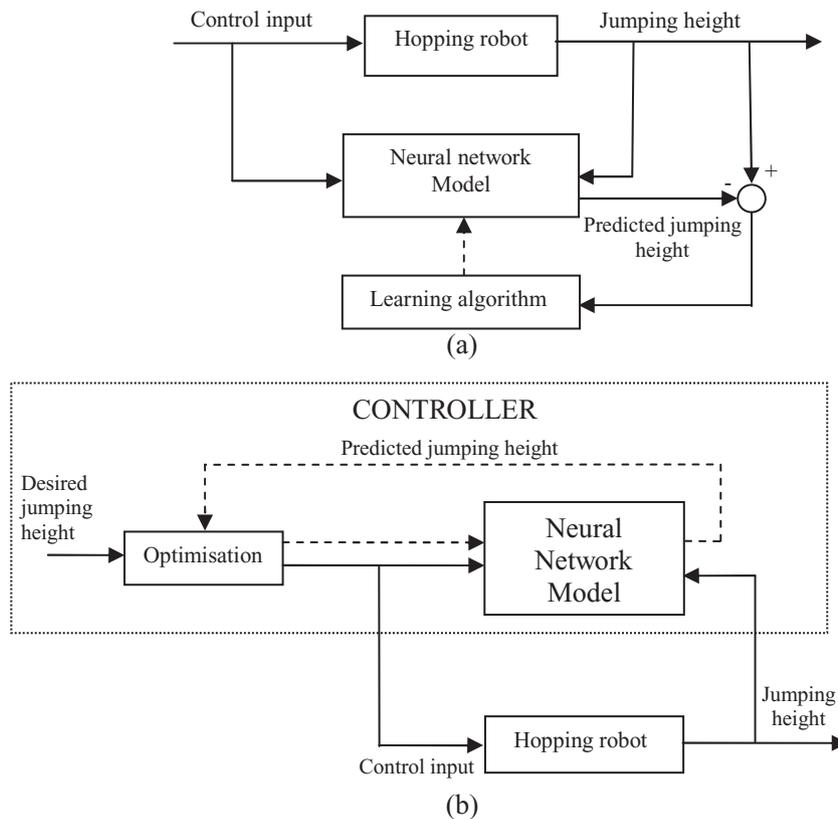


Figure 6. (a) The inverse dynamic calculation using a neural network; (b) ANN-based predictive control scheme.

from one cycle to another that should be accounted for in inverse dynamic calculations.

Generally speaking, in an artificial neural network, a new input to the network leads to an output similar to the correct output for input vectors used in training. This generalised property makes it possible to train a network on a representative set of input/target pairs and get good results without training the network on all possible input/output pairs. The constructed ANN model was a multi-layered Feedforward Back Propagation (FBP) network. The network was made of one hidden layer and an output layer. The output y in the FBP is related to the network parameters by the following equation:

$$y = f^2(LW^{2,1} f^1(IW^{1,1} p + b^1) + b^2), \quad (22)$$

where f^2 is the function being applied on layer 2 (i.e. the output layer in the proposed neural network) with weights $LW^{2,1}$. $IW^{1,1}$ is the input layer's weights, b^1 and b^2 are the bias terms and p is the input vector. The output y represents the jumping height of the hopper, namely y_2 .

To achieve the goal of controlling the jumping height of the hopping robot, a FBP network was trained to estimate the optimal value of the control variable, namely the stroke x at which the system should be pressurised, based on the user-defined desired maximum jumping height. After a certain number of trials, one 'pure linear' neuron for the output layer and fifteen 'tan sigmoid' neurons for the hidden layer were chosen in the network's structure. The

ANN was trained offline. The training and implementation processes were conducted through the following steps:

1. Training data assembly: To gather the training data, the hopper was made to jump by randomly changing the control input 'x' at which the system was pressurised.
2. Creation of the network object: The ANN object was constructed with 15 'tan-sigmoid' neurons in the middle layer, and one 'pure-linear' neuron in the output layer. The number of epochs, i.e. 1750 (trials), was selected through a trial and error procedure.
3. Network training: FBP was trained with the back propagation technique which involves execution of a series of backward iterations through the network to calculate the gradient of the ANN's performance function. In back propagation, the *Levenberg-Marquardt* optimisation algorithm was adopted for the training of the ANN instead of the *Quasi-Newton and Gradient Descent*, due to its better performance for moderate sized networks.
4. Subjecting the network to new testing inputs: The trained network was subjected to new input data via simulation to validate the performance of the trained network object.

The input/output layer weights and biases of the trained ANN object were derived and then implemented in the controller model built in SIMULINK in the form of an S-function.

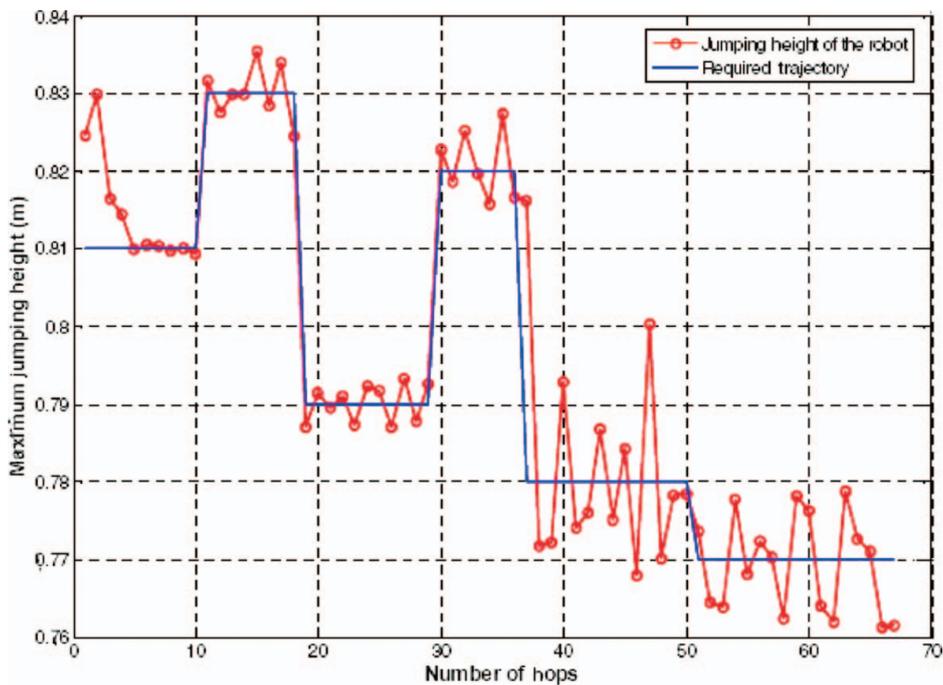


Figure 7. Experimental results of the Neuro-PD controller.

7. Experimental results and discussion

The PID controller adopted in simulation would make the steady state error very small at the cost of making the overall time response of the system slow. Besides, an anti-windup strategy must be adopted to avoid unwanted over/undershoots. To make the system's response faster without compromising the steady state error, a PD controller integrated with the trained ID-based ANN (called a Neuro-PD controller) was developed and implemented on a real system. Figure 7 shows the performance of the proposed controller when the robot was subjected to step changes in the desired jumping height on fly. The algorithm for implementing the proposed neuro-PD controller is as follows:

Whenever a change in the user-defined set point (i.e. a change in the desired trajectory) was detected, the ID-based ANN controller calculates the control variable (i.e. the stroke at which the pressurisation phase starts) to be adopted for the next hop. This control strategy is then applied within the next hopping cycle at the pressurisation phase. Consequently, the PD controller takes over in the next hopping cycle. Figure 7 shows a representation of more than 20 tests that were conducted to assess the performance of the controller. The same trend was observed for all tests. It is noteworthy that an integrator with an anti-windup strategy would be required to bring the steady-state error to zero, especially for large changes in the user-defined desired jumping height. One should note that the result given in Figure 7 depicts a representative test. However, the difference in numerous runs carried out under the same initial conditions remained very incremental. Figure 7 also depicts the robustness of the proposed control strategy in achieving the control objective when the system was subjected to seven step changes in the desired jumping height on fly. A maximum foot elevation of 20 centimetres was achieved. It can be increased by raising the pressure of air supply. This is comparable to the jumping height achieved by similar one-legged hopping machines using electromechanical actuators with a much higher power consumption.

8. Conclusions and future work

In this paper, the development of a comprehensive mechanistic model of a pneumatically actuated one-legged hopping robot was presented. The model was utilised to scale the pneumatic actuator and to synthesise different control strategies in a simulated environment prior to real implementation. The procedure for system identification through real-time open-loop tests was briefly discussed. Two sets of control variables, namely the duty cycle of the PWM signal applied to the on/off control valve at the touchdown phase and the stroke at which the pressurisation starts, were tested. Unwanted fast vibrations in the system's behaviour due to mechanical impacts were detected when utilising the

former control variable. P, PD, PID and Neuro-PD (inverse-dynamic based) control strategies were studied. The attributes of prediction and learning of an ANN were incorporated into the control design. The inverse dynamics concept was adopted to train a feedforward back propagation ANN offline. In order to increase the reliability and robustness of the controller, a well-tuned PD controller was integrated into the ANN, which resulted in a Neuro-PD controller. The main advantage of this approach was that the controller was able to move the robot from any jumping height to the desired height in a shorter time than that which could be achieved when using a stand-alone PD controller. This was particularly shown in a representative experimental test with six step changes in the desired jumping height on fly (see Figure 7). The performance of the proposed neuro-PD controller was found to be satisfactory in terms of the settling time and the steady state error. However, further tuning would be required to decrease the steady-state error in the system.

Our proposed control strategy does not address the online learning of the ANN controller, the addition of which would allow the network controller to be of an adaptive nature. The development of a next-generation hopping robot using an electromagnetic actuator along with a Magnetorheological-Fluid based motion damper for smoother jumps is presently under investigation. The current system can be further used as a test bed for implementing modal and/or supervisory control strategies.

References

- Buehler M, Cocosco A, Yamazaki K, Battaglia R. 1999. Stable open loop walking in quadruped robots with stick legs. Proceedings of the IEEE International Conference on Robotics and Automation. Detroit, MI, USA, Vol. 3. p. 2348–2353.
- Helferty J, Collins J, Kam M. 1989. A learning strategy for the control of a one-legged hopping machine. In Proceedings of the IEEE International Conference on Robotics and Automation; Scottsdale, USA. p. 1604–1609.
- Lebaudy A, Prosser J, Kam M. 1993. Control algorithms for a vertically constrained one-legged hopping machine. In Proceedings of 32nd IEEE Conference on Decision and Control. Vol. 3; San Antonio, TX, USA, 15–17 December 1993, p. 2688–2693.
- Mehrandezh M, Surgenor BW, Dean SRH. 1995. Jumping height control of an electrically actuated, one-legged hopping robot: modeling and simulation. Paper presented at the Proceedings of the 34th Conference on Decision & Control; New Orleans, LA, USA. p. 1016–1020.
- Naik K. 2006. Real-time active control of a pneumatically actuated one-legged hopping robot [M.Sc. thesis]. Canada: Faculty of Engineering, University of Regina.
- National Instruments 2008. <http://sine.ni.com/nips/cds/view/plang/en/nid/11913> (last visited July 2008).
- Norgren 2009. 24011 Series General Purpose Solenoid Valve. <http://www.norgren.com/usa/products/Section.asp?SPID=108> (last visited August 2009).
- Ogata K. 2002. Modern control engineering. 4th ed. Upper Saddle River (NJ): Prentice Hall.

- Papantoniou K. 1991. Electro-mechanical design for an electrically powered actively balanced one leg planar robot. Paper presented at the Proceedings of IEEE/RSJ International Workshop on Intelligent Robots and Systems. Vol. 3; Osaka, Japan. p. 1553–1560.
- Prosser J, Kam M. 1992. Vertical control for a mechanical model of the one-legged hopping machine. Paper presented at the Proceedings of IEEE 1st Conference on Control Applications; Daytona, USA. p. 136–141.
- Raibert MH. 1984. Hopping in legged systems, modeling and simulation for the two-dimensional one-legged case. *IEEE Trans SMC*. 14(3):451–463.
- Richer E, Hurmuzulu Y. 2000. A high performance pneumatic force actuator system Part-1 Nonlinear mathematical model. *ASME J Dyn Syst Meas Control*. 122(3):416–425.
- Sato A, Buehler M. 2004. A planar hopping robot with one actuator: design, simulation, and experimental results. *IEEE/RSJ 17th International Conference on Intelligent Robots and Systems (IROS)*; Sendai, Japan. p. 3540–3545.
- Sayyad A, Seth B, Seshu P. 2007. Single-legged hopping robotics research – A review. *Robotica*. 25(5):587–613.
- Sznair M, Damborg M. 1989. An adaptive controller for a one-legged mobile robot. *IEEE Trans Robot Autom*. 5(2):253–259.
- Werbos P. 1991. An overview of neural networks for control. *IEEE Contr Syst Mag*. 11(1):40–41.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

