JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 25, No. 2, March–April 2002

Engineering Notes

ENGINEERING NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes cannot exceed 6 manuscript pages and 3 figures; a page of text may be substituted for a figure and vice versa. After informal review by the editors, they may be published within a few months of the date of receipt. Style requirements are the same as for regular contributions (see inside back cover).

Adaptive Critic-Based Neural Networks for Agile Missile Control

Dongchen Han* and S. N. Balakrishnan†
University of Missouri–Rolla, Rolla, Missouri 65409-0050

I. Introduction

TO explore and extend the range of operations of air-to-air missiles, there have been studies in recent years with a completely different concept. That is, launch the missile as usual from the aircraft; however, the missile should be able to intercept a target in the rear hemisphere. The best emerging answer to execute this task is to use the aerodynamics and thrust to turn around the initial flight-path angle of zero to a final flight-path angle of 180 deg. (Every scenario can be considered as a subset of this set of extremes in flight-path angles.) In the design, as well as later phases of such a missile, there is a need to develop analysis tools to provide minimum time solutions. This problem falls under a class called free final time[1,2] problems in calculus of variations (optimal control), which is difficult to solve.

Several authors have used neural networks to solve problems optimally associated with control of nonlinear systems. For surveys of their works, see Hunt[3] and White and Sofge.[4] Almost all of the studies listed in these references use four types of neurocontrol: 1) supervised control, 2) direct inverse control, 3) neural adaptive control, and 4) backpropagation through time. A fifth and rarely studied class of controller has the most interesting structure. It is called an adaptive critic architecture.[5–8] Reasons for choosing this structure for formulating the optimal control problems are that this approach does not consist of external training as in some other controller designs, this is not an open-loop optimal controller but a feedback controller, and this approach preserves the same structure regardless of the problem (linear or nonlinear). Balakrishnan and Biega[5] have shown the usefulness of this architecture for infinite time linear problems. In this study, we present a general neural framework for the study of linear as well as nonlinear, finite time optimal control problems. Note that, in this study, we use computational adaptive critics in the sense that we use the neural networks to compute and embed optimal solutions without ramifications to strict reinforcement learning. The method discussed in this study determines an optimal control law for a system by successively adapting two networks, an action network for which the inputs are the states and the outputs are the values of control and a critic network, which has the states as inputs and the costates as outputs. A useful feature of this approach is that the same networks can be used to output optimal control for an entire range of initial conditions.

Presented as Paper 98-4495 at the AIAA Guidance, Navigation, and Control Conference, Boston, MA, 10–12 August 1998; received 27 December 1999; revision received 22 May 2001; accepted for publication 24 May 2001.

*Ph.D. Student, Department of Mechanical and Aerospace Engineering and Engineering Mechanics.

†Professor, Department of Mechanical and Aerospace Engineering and Engineering Mechanics.

II. System Model and Optimal Control

Equations of motion for a missile in a vertical plane are presented in this section. The minimum-time optimization problem is presented, the difficulties are pointed out, and a reformulation is made with the flight-path angle as the independent variable.

The nondimensional equations of motion (for a point mass) in a vertical plane are

$$M' = -S_w M^2 C_D - \sin\gamma + T_w \cos\alpha \tag{1}$$

$$\gamma' = (1/M)\left[S_w M^2 C_L + T_w \sin\alpha - \cos\gamma\right] \tag{2}$$

where prime denotes differentiation with respect to the nondimensional time τ.

The nondimensional parameters used in Eqs. (1) and (2) are

$$\tau = g/at, \qquad T_w = T/mg, \qquad S_w = \rho a^2 S/2mg, \qquad M = V/a$$

In these equations, M is the flight Mach number, γ the flight-path angle, α the aerodynamic angle of attack, T the solid rocket thrust, m the mass of the missile, S the reference aerodynamic area, V the speed of the missile, C_L the lift coefficient, C_D the drag coefficient, g the acceleration due to gravity, a the local speed of sound, ρ the atmospheric density, and t the flight time. Note that C_D and C_L are functions of angle of attack and Mach number.

The objective of the minimization process is to find the control (angle-of-attack) history to minimize the time taken by the missile to reverse its flight-path angle completely while the Mach number changes from an envelope of initial Mach numbers to a given final Mach number of 0.8.

Mathematically, this problem is stated as to find the control minimizing the cost function J:

$$J = \int_0^{t_f} \mathrm{d}t \tag{3}$$

with the constraints $\gamma(0) = 0$ deg, $M(0) \equiv$ given, $\gamma(t_f) = 180$ deg, and $M(t_f) = 0.8$. No general solution for this problem exists that can be used to generate optimal paths for flexible initial conditions.

We seek to provide such solutions using adaptive critic-based neural networks. To facilitate the solution using neural networks, the equations of motion are reformulated with flight-path angle as the independent variable. By doing so, we are able to obtain a fixed final condition in the independent variable. The transformed dynamic equations are

$$\frac{\mathrm{d}M}{\mathrm{d}\gamma} = \frac{\left(-S_w M^2 C_D - \sin\gamma + T_w \cos\alpha\right)M}{S_w M^2 C_L - \cos\gamma + T_w \sin\alpha} \tag{4}$$

$$\frac{\mathrm{d}t}{\mathrm{d}\gamma} = \frac{aM}{g\left(S_w M^2 C_L - \cos\gamma + T_w \sin\alpha\right)} \tag{5}$$

and the transformed cost function is

$$J = \int \left(\frac{aM}{g\left(S_w M^2 C_L \cos\gamma + T_w \sin\alpha\right)}\right) \mathrm{d}\gamma \tag{6}$$

with the limits on γ being 0 and π rad.

In this study, the final velocity is treated as a hard constraint. This means that the flight-path angle and the velocity constraints are met exactly at the final point. We express the dynamics and associated optimal control equations in a discrete form to use them with discrete feedforward neural networks. The system equations in a discrete form are

$$M_{k+1} = M_k + \frac{\left(-S_w M_k^2 C_{Dk} - \sin\gamma_k + T_{wk}\cos\alpha_k\right) M_k}{S_w M_k^2 C_{Lk} - \cos\gamma_k + T_{wk}\sin\alpha_k} \cdot \delta\gamma_k \tag{7}$$

$$t_{k+1} = t_k + \frac{a M_k \delta\gamma_k}{g\left(S_w M_k^2 C_{Lk} - \cos\gamma_k + T_{wk}\sin\alpha_k\right)} \tag{8}$$

The corresponding Hamiltonian equation (see Ref. 1) is

$$H_k = \frac{a M_k \cdot \delta\gamma_k}{g\left(S_w M_k^2 C_{Lk} - \cos\gamma_k + T_{wk}\sin\alpha_k\right)} + \lambda_{k+1} \times \left(M_k + \frac{\left(-S_w M_k^2 C_{Dk} - \sin\gamma_k + T_{wk}\cos\alpha_k\right) M_k}{S_w M_k^2 C_{Lk} - \cos\gamma_k + T_{wk}\sin\alpha_k} \cdot \delta\gamma_k\right) \tag{9}$$

For convenience, let us define an intermediate variable

$$\text{denk} = S_w M_k C_{Lk} - \cos\gamma_k + T_{wk}\sin\alpha_k$$

$$\frac{\partial \text{denk}}{\partial \alpha_k} = S_w M_k^2 \frac{\partial C_{Lk}}{\partial \alpha_k} + T_{wk}\cos\alpha_k$$

$$\frac{\partial \text{denk}}{\partial M_k} = 2 S_w M_k^2 C_{Lk} + S_w M_k^2 \frac{\partial C_{Lk}}{\partial M_k} \tag{10}$$

The costate equation is given in terms of denk as

$$\frac{\partial H_k}{\partial M_k} = \lambda_k = \frac{a \cdot \delta\gamma_k}{g \cdot \text{denk}} - \frac{a M_k \delta\gamma_k}{g \cdot \text{denk}^2} \cdot \frac{\partial\, \text{denk}}{\partial M_k} + \lambda_{k+1} \cdot \delta\gamma_k \times \frac{\left(-3 S_w M_k^2 C_{Dk} - S_w M_k^3 (\partial C_{Dk}/\partial M_k) - \sin\gamma_k + T_{wk}\cos\alpha_k\right)}{\text{denk}} + \lambda_{k+1} + \lambda_{k+1} \cdot \delta\gamma_k \frac{\left(S_w M_k^2 C_{Dk} + \sin\gamma_k - T_{wk}\cos\alpha_k\right) \cdot M_k}{\text{denk}^2} \times \frac{\partial\, \text{denk}}{\partial M_k} \tag{11}$$

Note that there is no boundary condition on λ because M is given at both ends.

Optimal control at each stage is obtained by setting the partial derivative of the Hamiltonian with respect to control to be zero at each stage. That is,

$$\frac{\partial H_k}{\partial \alpha_k} = 0$$

In an expanded form, this leads to

$$\frac{a}{g} \cdot \frac{\partial\, \text{denk}}{\partial \alpha_k} + \lambda_{k+1} \cdot \left(S_w M_k^2 \frac{\partial C_{Dk}}{\partial \alpha_k} + T_{wk}\sin\alpha_k\right) \cdot \text{denk} + \lambda_{k+1}\left(-S_w M_k^2 C_{Dk} - \sin\lambda_k + T_{wk}\cos\alpha_k\right) \cdot \frac{\partial\, \text{denk}}{\partial \alpha_k} = 0 \tag{12}$$

At every point we solve Eq. (12) using the Newton–Raphson method (see Ref. 1).

III. Development of Neural Network Solutions

The neural network solutions are obtained by using a pair of action and critic networks at each stage. At the last stage, the boundary conditions are used to generate the action and critic networks. For all of the lower stages until the first stage is reached, the immediate upper level critic helps in obtaining the lower level optimal controller (action) network.

Last Network

1) The final Mach number M_N is fixed at 0.8. Set $\Delta\gamma$. For random values of M_{N-1}, calculate α_{N-1} from the state propagation equation.

2) Use the optimality condition, $H_{\alpha N-1}(M_{N-1}, \lambda_N, \alpha_{N-1}) = 0$, where the subscript to H denotes partial differentiation, to solve for appropriate λ_N.

3) From the costate propagation equation, calculate λ_{N-1}.

4) Train two neural networks: The α_{N-1} network outputs α_{N-1} for different values of M_{N-1}, and the λ_{N-1} network outputs λ_{N-1} for different values of M_{N-1}. We have optimal α_{N-1} and λ_{N-1} now.

Other Networks

5) Assume different values of M_{N-2} and use a random neural network (or initialized with α_{N-1} network) called α_{N-2} network to output α_{N-2}. Use M_{N-2} and α_{N-2} to obtain M_{N-1}. Input M_{N-1} to λ_{N-1} network to get λ_{N-1}. Use M_{N-2} and λ_{N-1} in $H_{\alpha_{N-2}} = 0$ to solve for α_{N-2}. Use this α_{N-2} to correct the network. Continue

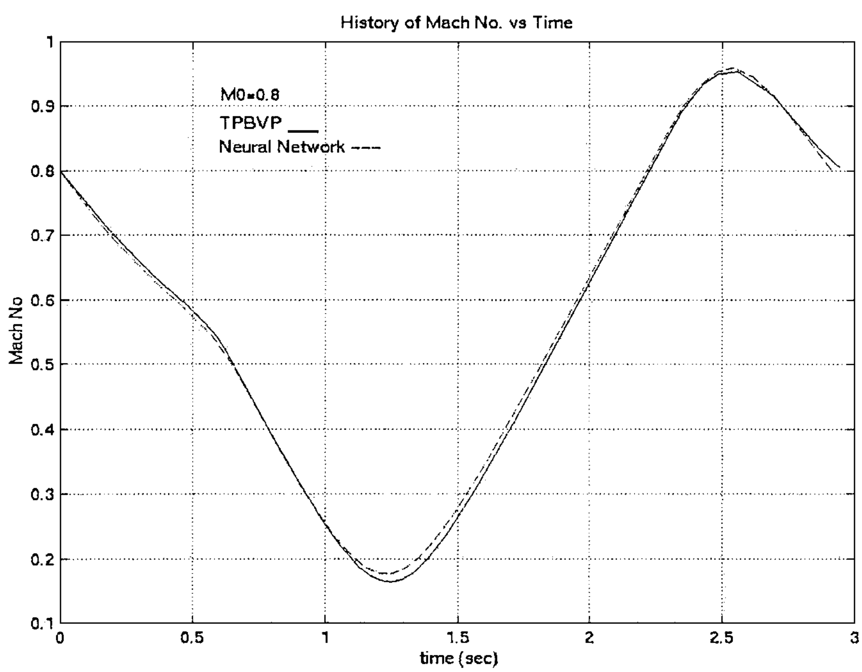

Fig. 1 Mach number comparison.

this process until the α_{N-2} network converges. This α_{N-2} network yields optimal α_{N-2}.

6) Using random M_{N-2} in the α_{N-2} network obtains optimal α_{N-2}. Use M_{N-2} and α_{N-2} to obtain M_{N-1}, and input to the λ_{N-1} network to generate λ_{N-1}. Use M_{N-2}, α_{N-2}, and λ_{N-1} in the costate equation to obtain optimal λ_{N-2}. Train the λ_{N-2} network with M_{N-2} as input. We have an λ_{N-2} network that yields optimal λ_{N-2}.

7) Repeat steps 5 and 6 with $K = N-1, N-2, \ldots, 0$, until we get α_0.

IV. Use of Networks in Real Time as Feedback Control

Start with any M_0 within the trained range. Because of the generalization capabilities of neural networks, we could go beyond the range to some extent. Use the α_0 neural network to find the optimal α, and integrate until γ_1 for the α_1 network is reached; use the M_1 values to find α_1 from the α_1 neural network, and integrate until γ_2 is reached, and so on, until γ_f is reached.

Note that the forward integration is done in terms of time [which is available as an incidental variable as a function of flight-path angle in Eq. (8)]. As a result, even though the network synthesis is done offline, the control is a feedback process based on current states.

V. Numerical Results

In this section, we present representative numerical results from the synthesis of the optimal control by using adaptive critic-based neural networks.

Tables of aerodynamic data of C_L and C_D variations with Mach number and angle of attack were provided by the U.S. Air Force. All of the networks in this study are feedforward networks; each has a three-layered structure with the first layer being a tangent sigmoid,

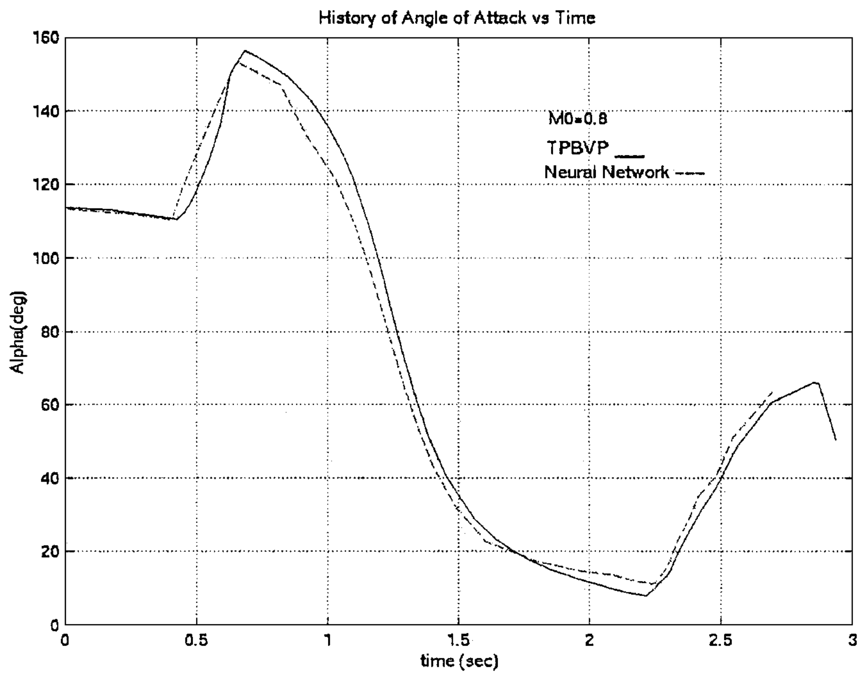

Fig. 2 Angle-of-attack comparison.

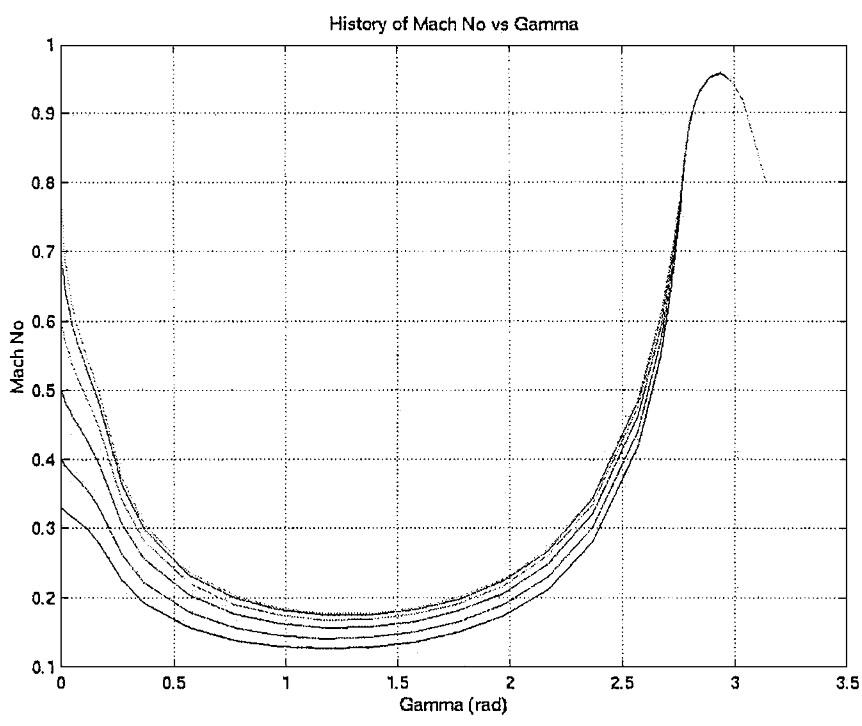

Fig. 3 Neural network solutions for various initial conditions.

the second layer a log sigmoid, and the third layer a linear network. They have nine neurons at each layer. The training method used in this study is Levenberg–Marquardt backpropagation method (see Ref. 4).

To demonstrate the optimality of the solutions resulting from the neural networks, we obtained optimal solutions to this minimum time problem using a shooting method[1] for one set of initial conditions. Flight Mach number histories from a shooting method, labeled two-point boundary value problem (TPBVP), and the neural networks are presented in Fig. 1. They are almost coincident showing that the neural solutions obtained with the cascade of controllers are (near) optimal. The initial Mach number for this selective example is 0.8. The associated angle-of-attack (control) histories are presented in Fig. 2. For most of the flight they are the same for both neural networks and the shooting method. The differences can be accounted for by the fact that the shooting method uses several more control steps than the 37 steps with neural networks. Although these plots establish the (near) optimality of the neurosolutions, the real advantage in using the adaptive critic approach is demonstrated in Fig. 3. For each trajectory with initial Mach number varying from 0.6 to 0.8, the final Mach number is 0.8. That is, the same cascade of neurocontollers is used to generate optimal control for an envelope of initial conditions. For all of the trajectories, the thrust profile was assumed to be constant. We carried out further numerical experiments to test the robustness of the controllers. We removed six controllers in the mid-flight-path region (which meant that some of the controls are held constant for longer periods) and plotted the missile trajectory. Even though the trajectory is less optimal, the lesser network configuration still delivers the missile to the exact final Mach number of 0.8.

Another advantage/flexibility of using these network controllers is that we can vary the initial flight-path angles for various Mach numbers. In these cases, all that is required to generate the optimal control to meet the final condition is to start from the network indexed with the initial flight-path angle and proceed to the next controllers as the flight-path angles changes. This is true by Bellman's principle of optimality (see Ref. 1), which states that, on an optimal path, trajectory from any intermediate stage to final stage is optimal for the given cost function.

VI. Conclusions

An adaptive critic-based neural network solution for a free final time problem associated with agile missile control has been developed. To our knowledge, there has been no other tool (other than dynamic programming) that provides such solutions. Also, the computational effort associated with the adaptive critic-based solutions is not prohibitive. This computational technique is fairly general and applicable to a wide variety of problems.

Acknowledgments

This research was supported by U.S. Air Force Grant F08630-96-1-0001 and National Science Foundation Grant ECS-9634127.

References

[1] Bryson, A. E., and Ho, Y., *Applied Optimal Control*, Hemisphere, New York, 1975, pp. 128–211.

[2] Stengel, R. F., *Optimal Control and Estimation*, Dover, New York, 1994, Chap. 3.

[3] Hunt, K. J., "Neural Networks for Controller Systems, A Survey," *Automatica*, Vol. 28, No. 6, 1992, pp. 1083–1112.

[4] White, D. A., and Sofge, D., *Handbook of Intelligent Control*, Van Nostrand Reinhold, New York, 1992, Chaps. 3, 5, 8, 12, 13.

[5] Balakrishnan, S. N., and Biega, V., "Adaptive Critic Based Neural Networks for Aircraft Optimal Control," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 4, 1996, pp. 893–898.

[6] Anderson, C. W., "Learning to Control an Inverted Pendulum Using Neural Networks," *IEEE Control Systems Magazine*, Vol. 9, No. 3, 1989, pp. 31–37.

[7] Barto, A. G., Sutton, R. S., and Anderson, C. W., "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-13, 1983, pp. 834–846.

[8] Werbos, P., "Neurocontrol and Supervised Learning; An Overview and Evaluation," *Handbook of Intelligent Control*, Van Nostrand Reinhold, New York, 1992, Chap. 3.

Jump Markovian-Based Control of Wing Deployment for an Uncrewed Air Vehicle

A. Stoica*
University "Polytechnica" of Bucharest, Bucharest RO 77206, Romania
and
I. Yaesh†
Israel Military Industries, 47100 Ramat-Hasharon, Israel

I. Introduction

AIR-LAUNCHED uncrewed air vehicles (UAVs) are often released with their wings folded to achieve clear and safe separation. The wings are deployed only when the UAV is required to begin a significant glide slope maneuver (see URL: http//vectorsite.tripod.com/avbomb9.html and URL: http//www.imi-israel.com). The ensuing abrupt change in aerodynamic forces and moments cause significant disturbances and result in a jump of the aerodynamic coefficients, thus leading to a transient in the angle of attack, which should be minimized to avoid loss of stability.

Note that the roll control problem also presents a challenge due to possible flow asymmetries that occur during the wing deployment transition period, resulting in large roll angle transients.

In the present Note, the focus is on the pitch control problem, where it is assumed that wing deployment is fast enough to assure that the system may be described by either wings-folded or wings-unfolded dynamics. This assumption stems from that speedy deployment mechanisms, able to complete their operation within $0.05-0.3$ s, are relatively low cost and simple to implement, for example, by a pneumatic piston. Slow deployment requires an appropriate servomechanism to achieve a smoothly controlled wing unfolding process. To avoid system complexity and the higher cost of such a servomechanism, a rapid wing deployment mechanism is preferred. Because the $0.05-0.3$ s wing deployment time is of the same order of magnitude as the aerodynamic short period, no intermediate wing positions need to be considered during the design process. Both force and moment coefficients are assumed, for design purposes, to undergo a jump at the instant of wing deployment, with the jump value depending on the instantaneous angle of attack. The angle of attack minimizing the lift during the jump phase differs from the one minimizing the pitch moment at the jump instant, and a compromise angle of attack of 0 deg is, therefore, chosen as the angle of attack for wing opening.

The UAV under consideration is equipped with a pitch rate sensor, a normal force accelerometer, and an angle-of-attack sensor. It is also equipped with a potentiometer indicating wing position. The zero-angle-of-attack design goal may also be interpreted as a 0-g requirement, thereby motivating the design of a 0-g commanded acceleration loop.

The aim of this Note is to present the design of a pitch acceleration loop whose task it is to minimize the possibly harmful effects of wing deployment. One possible approach involves embedding the plant into a convex polytope,[1] whose vertices contain the dynamic descriptions of the folded and unfolded wing aerodynamics,

Received 9 April 2001; revision received 13 September 2001; accepted for publication 21 November 2001.

*Professor, Faculty of Aerospace Engineering, Str. Splaiul Independentei No. 313; stoica@aeronet.propulsion.pub.ro.

†Head of Control Department, Advanced Systems Division, P.O.B. 1044/77; mlyaesh@isdnmail.co.il.