Direct method for real-time prototyping of optimal control

Yakimenko, Oleg A.

DIRECT METHOD FOR REAL-TIME PROTOTYPING OF OPTIMAL CONTROL

Oleg A. Yakimenko

Department of Mechanical and Astronautical Engineering
Naval Postgraduate School, Monterey, CA, USA

Abstract: This paper addresses the problem of real-time generation of near-optimal control for nonlinear plants. It advocates using the direct method of calculus of variations based on a priory approximation of the optimal solution by appropriate reference functions depending on several variable parameters and then applying inverse dynamics to determine the corresponding controls. The paper reviews general ideas of such an approach and provides with two detailed examples based on recent graduate students' projects. These include a standard inverted pendulum control and a more complex problem aimed on the development of algorithms for cooperative collision-free control of multiple agents. *Copyright © 2006 USTARTH*

Keywords: Optimal control, Real-time control, Aerospace and Aeronautical applications.

1. INTRODUCTION

It has been always desirable to have algorithms capable of solving different optimal control problems in real time. Unfortunately, classical indirect methods of calculus of variations (Bellman's dynamic programming, Pontrjagin's maximum principle) can handle online nothing but very simple problems (for these problems, double integrator for example, you might be able to obtain even an analytical solution, however it is quite difficult to obtain even an off-line numerical solution for the higher-fidelity models). That's why different techniques have been developed to either simplify (linearize) the problem or apply numerical algorithms providing a near-optimal rather than optimal solution (collocation method, method of differential inclusions, etc.). Once such a quasi-optimal solution is found in real time, it can then be tracked with conventional feedback controllers.

This paper advocates using one of the direct methods which has been proved to be able to work in real time already. In addition, this method is also quite easy to program and assures effective optimization within a compact set of feasible solutions.

This direct method of calculus of variations, similar to that of Ritz-Galerkin, applied to the problems of flight dynamics involving constraints on states and controls was originally developed in early 60s by Prof. Taranenko. Since then different modifications of Taranenko's method were developed by his followers for different specific applications (Taranenko, and Momdgi, 1986; Neljubov, 1986). A decade ago advances in computers allowed bringing Prof. Taranenko's ideas to a new level and developing algorithms for real-time on-board calculation of quasi-optimal trajectories (Yakimenko, 1998; Dobrokhodov, and Yakimenko, 1999; Alekhin and Yakimenko, 1999). These algorithms applied to combat vehicles and missiles were implemented and tested within the Pilot's Associate program onboard 5th-generation aircraft (Yakimenko, 2000a). Currently this direct method is being used for unmanned air vehicles (UAVs) to produce landing approach trajectories and assure flight deconfliction in case of maneuvering of multiple UAVs (Kaminer, *et al*., 2006).

Since key ideas of the onboard version of this direct method have been discussed in detail already (Yakimenko, 2000b), this paper focuses on applying its main ideas in different research projects at the Naval Postgraduate School, Monterey, CA. The simplicity of the direct method under discussion allows graduate students to advance through the theory, develop and eventually test their own real-time trajectory optimization schemes for different experimental setups in a variety of research laboratories at the department of Mechanical and Astronautical Engineering within one or two quarters.

While every project deserves a separate paper, the goal of the current paper is to show how the trajectory (state) optimization problem can be cast in terms of suggested direct method and reduced to the problem of constrained minimization of a scalar function of a few varied parameters. The paper starts with reviewing general ideas behind Taranenko's direct method and proceeds with two out of many different projects, walking through all computational burdens.

2. GENERAL IDEAS OF TARANENKO'S METHOD

To approximate the Cartesian coordinates of a vehicle and its speed (four states), Prof. Taranenko suggested using the following continuous, unequivocal and differentiable functions (Taranenko, and Momdgi, 1986)

$$x_i = x_{i0} + \frac{x_{if} - x_{i0}}{\tau_f - \tau_0}(\tau - \tau_0) + \Phi_i(\tau),\ i = \overline{1,4}. \quad (1)$$

Obviously these functions automatically satisfy both boundary conditions $x_i(\tau_0) = x_{i0}$ and $x_i(\tau_f) = x_{if}$ as long as $\Phi_i(\tau_0) = \Phi_i(\tau_f) = 0$. As an argument τ any monotonically changing parameter, e.g., time, path, full mechanical energy, etc. can be used.

Functions $\Phi_i(\tau)$ define the variety of accessible trajectories and their choice depends on the specific problem. For instance, the following functions and their combinations have been successfully used as $\Phi_i(\tau)$ in different applications: $\sum_{k=1}^{n} A_k \sin k\pi \frac{\tau - \tau_0}{\tau_f - \tau_0}$, $\sum_{k=1}^{n} A_k (\tau - \tau_0)^k (\tau - \tau_f)^k$, and $(\tau - \tau_0)^{m_1} (\tau - \tau_f)^{m_2}$.

In general, the more terms functions $\Phi_i(\tau)$ have, the more accurate (closer to the really optimal) solution can be found. Some of unknown coefficients A_k can be found using boundary conditions imposed on the first and higher-order derivatives $x_i^{(l)} = d^l x_i / d\tau^l$, $l = 1, 2, \ldots$. Other (free) coefficients become varied parameters.

Once the Cartesian coordinates and speed are defined using reference functions (1), the remaining states and controls are determined using inverse dynamics of original non-linear equations driving the system's dynamics. That is where Taranenko's direct method has a huge advantage over indirect methods. As known, in classical optimal control the problem is reduced to determining the trajectory with the given initial states and control time-histories by solving the Cauchy problem, i.e. integrating differential equations. Taranenko's direct method in general does not require solving the Cauchy problem. Instead, having the desired trajectory from the very beginning the inverse dynamics is applied to retrieve time-histories for all the controls.

We omit other features of Taranenko's method and jump into one of its modifications that employs polynomials as the reference functions (1).

3. COMPUTING COEFFICIENTS OF THE REFERENCE FUNCTIONS

Without loss of generality we further consider a single class of the reference functions, namely polynomials to show how unknown coefficients can be determined (Neljubov, 1986) and what can be done to assure an additional flexibility of trajectories.

Consider a typical case where you have to satisfy up to the second derivative of Cartesian coordinates at both ends of the trajectory: x_{i0}, x_{if}, x'_{i0}, x'_{if}, x''_{i0}, x''_{if}, $i = \overline{1,3}$ (since we haven't defined the argument of the reference functions, yet we use primes rather than dots to denote the derivatives). We also want the second derivate (which is proportional to acceleration or control actions) to be as smooth as a third-order polynomial

$$x''_i(\tau) = a_{i2} + a_{i3}\tau + a_{i4}\tau^2 + a_{i5}\tau^3 = \sum_{k=2}^{5} a_{ik}\tau^{k-2}. \quad (2)$$

Integrating equation (2) twice yields the following 4th- and 5th-order polynomials for the first derivative and ith coordinate

$$x'_i(\tau) = \sum_{k=1}^{5} \frac{a_{ik}\tau^{k-1}}{\max(1, k-1)} \text{ and}$$
$$x_i(\tau) = \sum_{k=0}^{5} \frac{a_{ik}\tau^{k}}{\max(1, k(k-1))}. \quad (3)$$

Now, all six coefficients a_{ik}, $k = \overline{0,5}$ can be defined from the following linear matrix equation:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & \tau_f & \frac{1}{2}\tau_f^2 & \frac{1}{6}\tau_f^3 & \frac{1}{12}\tau_f^4 & \frac{1}{20}\tau_f^5 \\ 0 & 1 & \tau_f & \frac{1}{2}\tau_f^2 & \frac{1}{3}\tau_f^3 & \frac{1}{4}\tau_f^4 \\ 0 & 0 & 1 & \tau_f & \tau_f^2 & \tau_f^3 \end{bmatrix} \begin{bmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ a_{i5} \end{bmatrix} = \begin{bmatrix} x_{i0} \\ x'_{i0} \\ x''_{i0} \\ x_{if} \\ x'_{if} \\ x''_{if} \end{bmatrix}. \quad (4)$$

In this particular case the only varied parameter is τ_f (since all coefficients are determined from the boundary conditions). Figure 1a demonstrates what happens when τ_f is varied. But what if in addition to τ_f some other coefficients can be varied? What if say the second derivative of x_1 at the final point, x''_{2f}, is not defined? Figure 1b provides with examples of such an enhanced flexibility of the reference functions.

To summarize, if an additional flexibility is needed – increase the order of polynomials and use the higher-order derivatives at both ends as additional varied parameters.

Now back to the argument of the reference functions. Assume $\tau \equiv t$. In this case the speed profile along the 3-dimentional trajectory is also determined (fixed) because $V = \sqrt{\dot{x}_1^2 + \dot{x}_2^2 + \dot{x}_3^2}$. For example we built a road where all cars are only allowed to move with a certain speed profile, which is absurd. The only way to avoid it is to use some abstract argument τ that relates to time via variable speed factor $\lambda(\tau) = d\tau / dt$. In this case we obtain an authority to vary the speed profile along the predetermined trajectory via varying the speed factor λ:

$$V = \lambda \sqrt{x_1'^2 + x_2'^2 + x_3'^2}\,. \qquad (5)$$

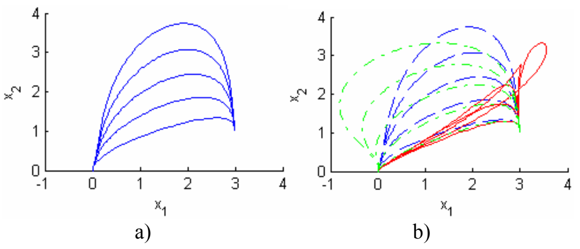


Fig. 1. Varying “free” parameters of the reference functions: $\tau_f = \text{var}$ (a), $\tau_f = \text{var}\,\&\, x''_{1f} = \text{var}$ (b).

With this let us further consider two optimization problems in detail.

4. EXAMPLES OF PROBLEM FORMULATIONS

This section presents two examples of casting the optimal control problem in terms of the direct method of calculus of variations to obtain a feasible quasi-optimal solution in real time. To address different features of implementation of fundamental ideas of the direct method, we start from the developing the real-time optimal control search routine for an inverted pendulum and proceed with the developing of coordinated control for multiple robots.

4.1 Inverted Pendulum

Physical System and Its Mathematical Model. The physical system is shown on Fig.2. By applying a torque to the horizontal disk, it is desirable to swing the pendulum from the inverted (down) position (or any other arbitrary angular position) to upright position and then keep it there.

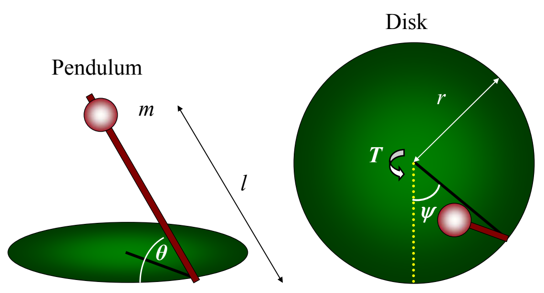


Fig. 2. Inverted pendulum.

The system of nonlinear equations driving system’s dynamics is shown below:

$$\begin{aligned}
\dot{\theta} &= \omega_\theta \\
\dot{\psi} &= \omega_\psi \\
\dot{\omega}_\theta &= f_1(\theta, \omega_\theta, \omega_\psi) + h_1(\theta)T \\
\dot{\omega}_\psi &= f_2(\theta, \omega_\theta, \omega_\psi) + h_2(\theta)T
\end{aligned} \qquad (6)$$

The four states are two attitude angles (defining a position of the pendulum and control disk, respectively) and corresponding angular velocities. The only control is the torque *T*. The nonlinear functions entering the equations (6) are defined as

$$\begin{aligned}
f_1 &= \frac{mlr\cos\theta\left[\left(I_x^p - I_y^p + ml^2\right)\dot{\psi}\dot{\theta}\sin\theta - mlr\dot{\theta}^2\sin\theta\right] + \frac{1}{2}\dot{\psi}^2\left(I_x^p - I_y^p + ml^2\right)\sin(2\theta) + mlg\sin\theta}{\left(I_z^p + ml\right)\left[I_y^d + mr^2 + \left(I_{x,p} + ml^2\right)\sin^2\theta + I_y^p\cos^2\theta\right] - \left(mlr\cos\theta\right)^2} \\
h_1 &= \frac{-mlr\cos\theta}{\left(I_z^p + ml\right)\left[I_y^p + mr^2 + \left(I_x^p + ml^2\right)\sin^2\theta + I_y^p\cos^2\theta\right] - \left(mlr\cos\theta\right)^2} \\
f_2 &= \frac{-\left(I_z^p + ml\right)\left[\left(I_x^p - I_y^p + ml^2\right)\dot{\psi}\dot{\theta}\sin\theta + mlr\dot{\theta}^2\sin\theta\right] - mlr\cos\theta\left[\frac{1}{2}\dot{\psi}^2\left(I_x^p - I_y^p + ml^2\right)\sin(2\theta) + mlg\sin\theta\right]}{\left(I_z^p + ml\right)\left[I_y^d + mr^2 + \left(I_x^p + ml^2\right)\sin^2\theta + I_y^p\cos^2\theta\right] - \left(mlr\cos\theta\right)^2} \\
h_2 &= -\frac{I_z^p + ml}{mlr\cos\theta}h_1
\end{aligned} \qquad (7)$$

and depend on some of the states and mass-geometric data of the system defined by the vector

$$\mathbf{a} = \left[m, l, r, I_y^d, I_x^p, I_y^p, I_z^p\right]^T, \ \mathbf{a} \in A^7 \subset E^7. \qquad (8)$$

It is desirable to satisfy the following set of boundary conditions at the initial and final points

$$
\begin{array}{ll}
\theta(t_0)=\theta_0 & \theta(t_f)=2\pi n,\ \ where\ n=0,\pm1,\dots \\
\psi(t_0)=\psi_0 & \psi(t_f)=0 \\
\dot\theta(t_0)=\dot\theta_0=\omega_{\theta 0} & \dot\theta(t_f)=\omega_{\theta f}=0 \\
\dot\psi(t_0)=\dot\psi_0=\omega_{\psi 0} & \dot\psi(t_f)=\omega_{\psi f}=0 \\
\ddot\theta(t_0)=\ddot\theta_0 & \ddot\theta(t_f)=0 \\
\ddot\psi(t_0)=\ddot\psi_0 & \ddot\psi(t_f)=0
\end{array}
$$

as well as obey the mechanical constraints on the control disk attitude $|\psi(t)|\le\psi_{\max}$ and control torque $|T(t)|\le T_{\max}$.

The performance index J is the time needed to bring the pendulum to the upright position. Of course the optimal control (and attitude time profiles) depends on the vector of system's parameters $\mathbf{a}$ (8).

Key Ideas for the Numerical Solution. Following the basic idea of the direct method, it is first needed to separate the trajectory (attitudes of the pendulum and control disk) from the velocity (their angular velocities in this case). In order to do so, an independent argument τ (virtual arc) needs to be introduced. Using the speed factor $\lambda=\dfrac{d\tau}{dt}$ the original system (6) becomes

$$
\begin{aligned}
\theta' &= \lambda^{-1}\omega_\theta \\
\psi' &= \lambda^{-1}\omega_\psi \\
\omega_\theta' &= \lambda^{-1}f_1(\theta,\omega_\theta,\omega_\psi)+\lambda^{-1}h_1(\theta)T \\
\omega_\psi' &= \lambda^{-1}f_2(\theta,\omega_\theta,\omega_\psi)+\lambda^{-1}h_2(\theta)T
\end{aligned}
\tag{9}
$$

Now, suggested by the optimal control theory (since the torque T enters the Hamiltonian of the system linearly) we assume the control-arc profile to be bang-bang as shown on Fig.3.

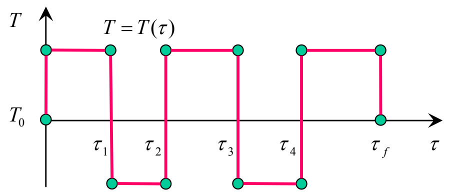
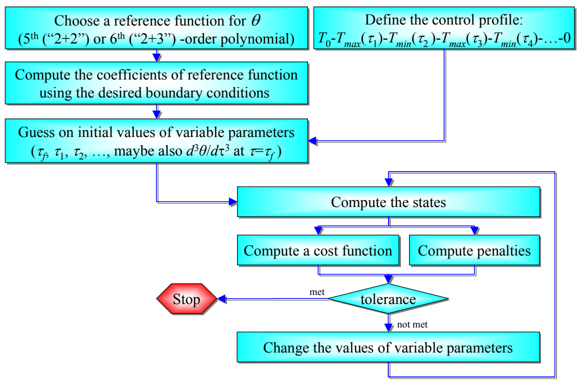


Fig. 3. Torque-arc profile.

This arc profile is completely defined by a series of switching points. Although no hint about the number of switching points exists (the only knowledge we have is that for a linearized system the number of switching points would not exceed three), we assume that we can establish the optimal control using some finite number of them, say four points as shown on Fig.3. Later we could adjust the number of points if necessary (we should probably need to minimize the number of switches as well).

Finally, we establish some reference function $P_\theta(\tau)$ for the attitude angle θ. For example we may go with the polynomial of as low as 5th order (since we need to satisfy up to the second derivative of the attitude angle at both ends of the trajectory). The coefficients of this polynomial are defined by the given boundary conditions (converted to argument τ) as explained in Section 3. (For more flexibility we may need to increase the order of the polynomial so that the third derivatives θ_0''' and/or θ_f''' become additional varied parameters.)

Now, the optimization routine may be established as shown on Fig.4.



Choose a reference function for θ (5th ("2+2") or 6th ("2+3") -order polynomial)
Define the control profile: T_0-$T_{max}(\tau_1)$-$T_{min}(\tau_2)$-$T_{max}(\tau_3)$-$T_{min}(\tau_4)$-…-0
Compute the coefficients of reference function using the desired boundary conditions
Guess on initial values of variable parameters (τ_f, τ_1, τ_2, …, maybe also $d^3\theta/d\tau^3$ at $\tau=\tau_f$)
Compute the states
Compute a cost function
Compute penalties
Stop
met
tolerance
not met
Change the values of variable parameters

Fig. 4. Optimization flow chart.

Given the boundary conditions we first define the reference function for the angle θ and compute all coefficients using these boundary conditions (and initial guesses on θ_0''' and/or θ_f''' in case higher than 5th-order polynomial was employed for more flexibility). We also establish a bang-bang torque-arc profile defined by several switching points. These switching points, τ_i, $i=\overline{1,4}$, along with the length of the virtual arc τ_f (and possibly the values of higher-order derivatives of the angle θ at initial and/or final points) form the vector of variable parameters $\boldsymbol{\Xi}$.

Next, we numerically solve the problem (integrating some of state equations where it is inevitable and applying inverse dynamics everywhere else). Then, we estimate the performance index J and compound the aggregated penalty Δ (caused by the fact that not all constraints are necessarily met and that the boundary conditions for the states that were integrated are not satisfied).

Now, we apply any standard nonlinear constrained minimization routine to minimize the performance index keeping the penalty within a certain tolerance ε

$$
\min_{\Xi} J\Big|_{\Delta\le\varepsilon}\,. \tag{10}
$$

Computation of States, Cost Function and Penalty. We start from dividing the virtual arc τ_f onto N–1 equal pieces $\Delta\tau=\dfrac{\tau_f}{N-1}$ so that we have N equidistant (along τ_f) nodes $j=\overline{1,N}$. All the states and the torque at the first point $j=1$ (corresponding to $\tau_1\equiv\tau_0=0$) are defined. Additionally we define $\lambda_1=1$.

Now, for each of the subsequent N–1 nodes $j=\overline{2,N}$ we do the following. We compute the current value of angle θ using the corresponding polynomial: $\theta_j = P_\theta(\tau_j)$. Next, knowing the torque from the predetermined torque-arc profile, $T_{j-1}=T(\tau_{j-1})$, we integrate the third equation of the system (9)

$$\omega_{\theta,j}=\lambda_{j-1}^{-1}\left(f_1(\theta_{j-1},\omega_{\theta,j-1},\omega_{\psi,j-1})+h_1(\theta_{j-1})T_{j-1}\right)\Delta\tau \ .(11)$$

Knowing the change in the pendulum angle and the angular velocity at each node allows computing the elapsed time

$$\Delta t_{j-1}=2\frac{\theta_j-\theta_{j-1}}{\omega_{\theta,j}+\omega_{\theta,j-1}} \quad (12)$$

and therefore the current value of the speed factor

$$\lambda_j=\frac{\Delta\tau}{\Delta t_{j-1}}. \quad (13)$$

The current time then equals to

$$t_j=t_{j-1}+\Delta t_{j-1} \quad (t_1\equiv t_0{=}0)\,. \quad (14)$$

Finally, we numerically integrate two remaining equations of the system (6) to get the current values for the remaining states

$$\begin{aligned}\omega_{\psi,j}&=\left(f_2(\theta_{j-1},\omega_{\theta,j-1},\omega_{\psi,j-1})+h_2(\theta_{j-1})T_{j-1}\right)\Delta t_{j-1}\\ \psi_j&=\omega_{\psi,j}\Delta t_{j-1}\end{aligned} \quad (15)$$

Once all states are computed, we may estimate the performance index which happens to be $J=t_N$ and form the penalty as

$$\Delta=\left[w,1-w\right]\begin{bmatrix}\max\limits_j\left(0;\left(\left|\psi_j\right|-\psi_{\max}\right)^2\right)\\ T_{\max}^{-1}\max\limits_j\left(0;\left(\left|T_j\right|-T_{\max}\right)^2\right)\end{bmatrix}, \quad (16)$$

where w is the weighting coefficient.

4.2 Cooperative Control of Multiple Robots

Experimental Setup and Mathematical Model. Experimental setup shown on Fig.5 includes three robots moving along a frictionless bounded horizontal arena. Each robot has three degrees of freedom (two translational and one rotational about the vertical axis) and is controlled by a reaction wheel and a thruster mounted atop of it. The robots should maneuver themselves collisions-free into a commanded final position defined relative to each other.

The system of nonlinear equations driving each robot's dynamics ($i=\overline{1,3}$) is given below:

$$\begin{aligned}\dot{x}^i&=u^i\\ \dot{y}^i&=v^i\\ \ddot{x}^i&=\dot{u}^i=F^i\cos(\psi^i+\alpha^i)\\ \ddot{y}^i&=\dot{v}^i=F^i\sin(\psi^i+\alpha^i)\\ \dot{\psi}^i&=\omega^i\\ \ddot{\psi}^i&=\dot{\omega}^i=T^i\\ \dot{\alpha}^i&=\delta^i\end{aligned} \quad (17)$$

The seven states per robot are its x and y coordinates, x and y, components of its linear velocity, u and v, respectively, the attitude angle ψ (defining robot's orientation with respect to the axis Ox), the angular velocity ω controlled by the reaction wheel, and the angle α defining the direction of thrust with respect to robot front. Three available controls (per robot) are the magnitude of its linear acceleration $F^i=\frac{Thrust^i}{m^i}$ ($0\le F^i\le F^i_{\max}$), the control input δ affecting orientation of the thrust and the angular acceleration $T^i=\frac{Torque^i}{I^i}$ ($\left|T^i\right|\le T^i_{\max}$).

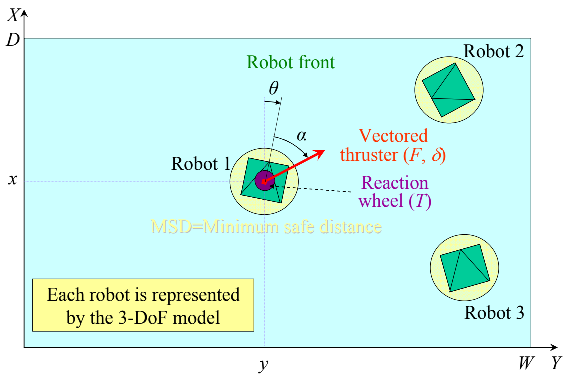


Fig. 5. Three robots on the frictionless arena.

While maneuvering, all robots ($i=\overline{1,3}$) must obey the geometrical constraints of the arena

$$\begin{aligned}&0.5MSD\le x^i(t)\le D-0.5MSD\,,\\ &0.5MSD\le y^i(t)\le W-0.5MSD\,,\ t\in\left[t_0,t_f\right]\end{aligned} \quad (18)$$

(where MSD stands for minimum safe distance between two robots and is equal to the diameter of the circles drawn on Fig.5 around each robot), and avoid collisions with other robots

$$\begin{aligned}&\left(x^i(t)-x^k(t)\right)^2+\left(y^i(t)-y^k(t)\right)^2-MSD^2\ge 0\\ &\forall\ i,k=\overline{1,3},\ i\ne k\,,\ t\in\left[t_0,t_f\right]\end{aligned} \quad (19)$$

It is required to satisfy the following sets of boundary conditions per each robot ($i=\overline{1,3}$):

$$
\begin{array}{ll}
x^i(t_0)=x_0^i & x^i(t_f^i)=x_f^i \\
y^i(t_0)=y_0^i & y^i(t_f^i)=y_f^i \\
\psi^i(t_0)=\psi_0^i & \psi^i(t_f^i)=\psi_f^i \\
\dot{x}^i(t_0)=u_0^i & \dot{x}^i(t_f^i)=u_f^i \\
\dot{y}^i(t_0)=v_0^i & \dot{y}^i(t_f^i)=v_f^i \\
\dot{\psi}^i(t_0)=\omega_0^i & \dot{\psi}^i(t_f^i)=\omega_f^i \\
\ddot{x}^i(t_0)=F_0^i\cos(\psi_0^i+\alpha_0^i) & \ddot{x}^i(t_f^i)=F_f^i\cos(\psi_f^i+\alpha_f^i) \\
\ddot{y}^i(t_0)=F_0^i\sin(\psi_0^i+\alpha_0^i) & \ddot{y}^i(t_f^i)=F_f^i\sin(\psi_f^i+\alpha_f^i) \\
\ddot{\psi}^i(t_0)=T_0^i & \ddot{\psi}^i(t_f^i)=T_f^i
\end{array}
$$

In general, the performance index includes three appropriately weighted terms. The first one, t_f^1, assures minimum transition time for the first robot, the second one, $\left|t_f^2-t_f^1-\Delta_t\right|+\left|t_f^3-t_f^2-\Delta_t\right|$, guarantees sequential (Δ_t-second apart) joining the final formation, and the third one, $\sum_{r=1}^{3}\int_{t_0}^{t_f^i}F^i dt$, takes care of minimizing overall gas consumption to produce thrust.

<u>Suggested Solution</u>. Again, the first step is to introduce an independent argument τ^i for each robot ($i=\overline{1,3}$) and using the corresponding speed factors λ^i (different for each robot) rewrite the original system (17) as

$$
\begin{aligned}
x'^i &= u^i/\lambda^i \\
y'^i &= v^i/\lambda^i \\
u'^i &= F^i\cos(\psi^i+\alpha^i)/\lambda^i \\
v'^i &= F^i\sin(\psi^i+\alpha^i)/\lambda^i \\
\psi^{i\prime} &= \omega^i/\lambda^i \\
\omega^{i\prime} &= T^i/\lambda^i \\
\alpha^{i\prime} &= \delta^i/\lambda^i
\end{aligned}
\tag{20}
$$

The second step is to establish three reference functions (per robot): for coordinates x^i and y^i, as well as for the attitude angle ψ^i: $P_x^i(\tau^i)$, $P_y^i(\tau^i)$ and $P_\psi^i(\tau^i)$, respectively ($i=\overline{1,3}$). As defined by the number of boundary conditions, the minimum order of approximating polynomials is five. For this specific problem the additional flexibility will definitely be needed to allow avoiding collisions. Thus, to be able to vary the third derivative of x^i, y^i and ψ^i, $i=\overline{1,3}$ at both ends, the order of polynomials should be increased by two.

The optimization routine looks pretty much the same as on Fig.4 except there is no predetermined control-arc histories established this time.

Given the boundary conditions we first determine nine reference functions $P_x^i(\tau^i)$, $P_y^i(\tau^i)$ and $P_\psi^i(\tau^i)$, $i=\overline{1,3}$ and compute their coefficients using given boundary conditions and initial guesses on the third derivatives $x_0^{i\prime\prime\prime}$, $x_f^{i\prime\prime\prime}$, $y_0^{i\prime\prime\prime}$, $y_f^{i\prime\prime\prime}$, $\psi_0^{i\prime\prime\prime}$, $\psi_f^{i\prime\prime\prime}$, $i=\overline{1,3}$. These variables along with the lengths of three virtual arcs τ_f^i form the vector of variable parameters Ξ. Next, applying inverse dynamics we numerically solve the problem for the remaining states. Then, we estimate the performance index J and compound the aggregated penalty Δ.

As in the previous example we apply any standard nonlinear constrained minimization routine to minimize the performance index keeping the penalty within the certain tolerance (10).

<u>Inverse Dynamics</u>. Again we start from dividing each virtual arc τ_f^i ($i=\overline{1,3}$) onto N–1 equal pieces $\Delta\tau^i=\dfrac{\tau_f^i}{N-1}$ so that we have N equidistant nodes $j=\overline{1,N}$ along each virtual arc. For each robot all states at the first point $j=1$ (corresponding to $\tau_1^i=\tau_0=0$) are defined. Additionally we define $\lambda_1^i=1$, $i=\overline{1,3}$.

For each of the subsequent N–1 nodes $j=\overline{2,N}$ we compute the current values of robots' coordinates and attitudes using corresponding polynomials: $x_j^i=P_x^i(\tau_j^i)$, $y_j^i=P_y^i(\tau_j^i)$ and $\psi_j^i=P_\psi^i(\tau_j^i)$, $i=\overline{1,3}$. Then, using the inverse dynamics for the first four equations of the system (20) we determine the sum of angles ψ_j^i and α_j^i, and current control acceleration

$$
\psi_j^i+\alpha_j^i=\arctan\left(\frac{y_j^{i\prime\prime}}{x_j^{i\prime\prime}}\right),\quad F_j^i=\lambda_j^i\sqrt{x_j^{i\prime\prime 2}+y_j^{i\prime\prime 2}}\,. \tag{21}
$$

Inverting the last equation of the system (20) and using the first of two equations (21) we obtain the second control

$$
\delta_j^i=\lambda_j^i\alpha_j^{i\prime}=\lambda_j^i\left(\frac{x_j^{i\prime\prime\prime}y_j^{i\prime\prime}-x_j^{\prime\prime i}y_j^{i\prime\prime\prime}}{y_j^{i\prime\prime}}\cos^2\psi_j^i-\psi_j^{i\prime}\right). \tag{22}
$$

From the first two equations of the system (20) we define the current speed

$$
V_j^i=\sqrt{u_j^{i2}+v_j^{i2}}\text{, where } u_j^i=\lambda_j^i x_j^{i\prime},\ v_j^i=\lambda_j^i y_j^{i\prime} \tag{23}
$$

and therefore may proceed with determining the elapsed time for each robot

$$
\Delta t_{j-1}^i=2\frac{\sqrt{\left(x_j^i-x_{j-1}^i\right)^2+\left(y_j^i-y_{j-1}^i\right)^2}}{V_j^i+V_{j-1}^i}\,. \tag{24}
$$

Now, the current values of the speed factor are given by

$$\lambda_j^i = \frac{\Delta\tau^i}{\Delta t_{j-1}^i} \qquad (25)$$

and the current time for each robot is defined as

$$t_j^i = t_{j-1}^i + \Delta t_{j-1}^i \quad (t_1^i{=}0)\,. \qquad (26)$$

Finally, we inverse the equations for robots' attitude to get the third control

$$\omega_j^i = 2\frac{\psi_j^i - \psi_{j-1}^i}{\Delta t_{j-1}^i} - \omega_{j-1}^i \text{ and } T_j^i = \frac{\omega_j^i - \omega_{j-1}^i}{\Delta t_{j-1}^i}. \quad (27)$$

Once all states along the trajectories are computed, we determine the performance index (employing the vector of weighting coefficients **w** ($\sum_{h=1}^{3} w_h = 1$))

$$J = \left[w_1, w_2, w_3\right]\begin{bmatrix} t_f^1 \\ \left|t_f^2 - t_f^1 - \Delta_t\right| + \left|t_f^3 - t_f^2 - \Delta_t\right| \\ \sum_{r=1}^{3}\sum_{j=0}^{N-1} F^i \Delta t_j^r \end{bmatrix} \quad (28)$$

and form the aggregate penalty using an appropriate four-component vector of weighting coefficients **k** ($\sum_{q=1}^{4} k_q = 1$):

$$\Delta = \left[k_1, k_2, k_3, k_4, k_4, k_4\right]\begin{bmatrix} \sum_{i=1}^{3}\left(\max_j\left(0; F_j^i - F_{\max}^i\right)\right)^2 \\ \sum_{i=1}^{3}\left(\max_j\left(0; \left|T_j^i\right| - T_{\max}^i\right)\right)^2 \\ \sum_{i=1}^{3}\left(\left(\max_j\left(0; \left|x_j^i - \frac{D}{2}\right| - \frac{D}{2} + MSD\right)\right)^2 + \left(\max_j\left(0; \left|y_j^i - \frac{W}{2}\right| - \frac{W}{2} + MSD\right)\right)^2\right) \\ \max_j\left(0; MSD^2 - \left(x^1(t_j^*) - x^2(t_j^*)\right)^2 - \left(y^1(t_j^*) - y^2(t_j^*)\right)^2\right), \; * = \arg\min_{i=1,2}(t_f^i) \\ \max_j\left(0; MSD^2 - \left(x^1(t_j^*) - x^3(t_j^*)\right)^2 - \left(y^1(t_j^*) - y^3(t_j^*)\right)^2\right), \; * = \arg\min_{i=1,3}(t_f^i) \\ \max_j\left(0; MSD^2 - \left(x^2(t_j^*) - x^3(t_j^*)\right)^2 - \left(y^2(t_j^*) - y^3(t_j^*)\right)^2\right), \; * = \arg\min_{i=2,3}(t_f^i) \end{bmatrix}. \quad (29)$$

Note that the last three terms in the compound penalty (29) are quite tricky because robots' coordinates have to be interpolated so that they correspond to the same instants of time.

5. CONCLUSION

The proposed direct method of calculus of variations provides explicit numerical schemes for robust real-time prototyping of near-optimal solutions for any complex system described by the ordinary nonlinear differential equations and employing any complex performance index. Among other advantages of this approach over other direct and indirect methods there are: feasibility of solutions with a priori unconditional satisfaction of all boundary conditions, a small number of variable parameters, and the absence of "wild" trajectories during optimization. The simplicity of this approach and excellent convergence properties allow implementing it easily in different real-time control projects by students of engineering curricula.

REFERENCES

Taranenko, V. and V. Momdgi (1986). *Direct Variational Method in Boundary Problems of Flight Dynamics*, Mashinostroenie, Moscow.

Neljubov, A. (1986). *Mathematical Methods of Computation of Combat, Takeoff and Landing Maneuvers for the Aircraft with Thrust Vectoring Capability*, Air Force Engineering Academy Press, Moscow.

Yakimenko, O.A. (1998). Simplified Modification of the Direct Variational Method for Onboard Solution of Optimization Boundary Problems for Flight Vehicle Trajectories, *Journal of Computer and Systems Sciences International*, 37(**3**), 405-415.

Dobrokhodov, V.N. and O.A. Yakimenko (1999). Synthesis of Trajectorial Control Algorithms at the Stage of Rendezvous of an Airplane with a Maneuvering Object, *Journal of Computer and Systems Sciences International*, 38(**2**), 262-277.

Alekhin, D.V. and O.A. Yakimenko (1999). Synthesis of Optimization Algorithm for Route Trajectory by the Direct Variational Method, *Journal of Computer and Systems Sciences International*, 38(**4**), 650-666.

Yakimenko, O. (2000a). Rapid Onboard Prototyping of Near-Optimal Spatial Trajectories for Pilot's Associate. *Proceedings of the 22nd International Congress of Aeronautical Sciences*, Harrogate, United Kingdom, August 27 – September 1.

Kaminer, I.I., O.A. Yakimenko and A.M. Pascoal (2006). Coordinated Control of Multiple UAVs for Time-Critical Applications. *Proceedings of the 27th IEEE Aerospace Conference*, Big Sky, Montana, March 4-11.

Yakimenko, O. (2000b). Direct Method for Rapid Prototyping of Near-Optimal Aircraft Trajectories. *AIAA Journal of Guidance, Control, and Dynamics*, 23(**5**), 865-875.