Comments on a Numerical Procedure for the Solution of Differential Games

G. J. OLSDER

Abstract—Two examples are given which show that one should be very cautious when using an appealing numerical approach to solve differential games or optimal control problems with more than one control variable. This numerical procedure essentially reduces the problem to a set of subproblems in which the optimization is considered with one control variable only.

A differential game with two players is given. Player $i$ wants to maximize criterion $J_i$, $i = 1, 2$. The game may be zero-sum ($J_1 = -J_2$), but this is not necessary. Player $i$ influences the game via the control $u_i(t)$, ($i = 1, 2$), which can be a scalar as well as a vector function.

In general, numerical procedures to obtain the optimal solution (Nash-solution) are rather complicated. An appealing approach, recommended for instance in [1], is the following one.

Step 1: Choose an initial set of controls $u_i^{(0)}(t)$, $u_j^{(0)}(t)$.

Step 2: Keeping $u_i^{(0)}(t)$ fixed, solve the optimization problem $\max u_j$ $J_j$,... As a result $u_j^{(1)}(t)$ is obtained.

Step 3: Keeping $u_i^{(1)}(t)$ fixed, solve $\max u_j$ $J_j$. As a result $u_j^{(1)}(t)$ is obtained.

Step 4: Return to Step 2, using $u_j^{(1)}(t), u_j^{(0)}(t)$ as the new reference solution.

Step 5: The procedure is continued until changes in $J$ and/or $u_i(t)$, from iteration to iteration, fall below a prescribed small difference.

This procedure, however, does not necessarily converge to the optimal solution. Even when the initial controls are arbitrarily close to the optimal ones, convergence does not always occur. This is rather obvious in programming problems and should be just as obvious in dynamical control problems, but is often overlooked because of the complexity of the problem, which, however, is of a different kind.

Two examples will be given; the first one with $J_1 = J_2$, the other one with $J_1 = J_2$, for which the numerical procedure proposed does not lead to the optimal solution.

Example 1: Consider the function $f(u_1, u_2) = -u_1 - u_2 + 2u_1u_2 - 1$, defined on the square $[0, 1] \times [0, 1]$. It is easily verified that

$$\max_{u_1 \in [0,1]} \min_{u_2 \in [0,1]} f(u_1, u_2) = \min_{u_2 \in [0,1]} \max_{u_1 \in [0,1]} f(u_1, u_2)$$

and that this is achieved for $u_1 = u_2 = \frac{1}{2}$. If the numerical procedure proposed is applied to this programming problem we get the following sequence of solutions (Player 1 wants to maximize $f$, Player 2 wants to minimize $f$):

$$u_1^{(0)} = 0, u_1^{(1)} = 1, u_1^{(2)} = 1, u_2^{(3)} = 0, u_2^{(3)} = 1, \ldots$$

which clearly never converges.

A differential game with the same features is

$$\frac{dx}{dt} = -u_1 - u_2 + 2u_1u_2 + 1, \quad x(0) = 0,$$

with constraints $|u_1(t)| < 1, |u_2(t)| < 1, \quad J_1 = x(1), \quad J_2 = -x(1)$.

If the initial solution for $u_i(t)$ is $u_i^{(0)}(t) \equiv 0, 0 < t < 1$, then the numerical procedure keeps jumping and will never tend to the optimal solution $u_i(t) = u_i(t) = 0, 0 < t < 1$.

If for the initial solution $u_i(t)$ a constant function arbitrarily close to $u_i(t)$, but different from $\frac{1}{2}$ is chosen, then the method does not converge either.

Example 2: Consider the function $g(u_1, u_2)$, defined by

$$g(u_1, u_2) = \begin{cases} 
-1 / (u_1 + 1), & 0 < u_1 < u_2 < 1, \\
-1 / (u_2 + 1), & 0 < u_2 < u_1 < 1.
\end{cases}$$

It is easily proven that

$$\max_{u_2 \in [0,1]} \max_{u_1 \in [0,1]} g(u_1, u_2) = 1$$

and that this maximum is achieved for $u_1 = u_2 = 0$. Using the numerical procedure described we get, if $u_i^{(0)} = \alpha$, where $\alpha$ is some number $\in [0,1]:$

$$u_1^{(0)} = \alpha, \quad u_1^{(1)} = \alpha, \quad u_1^{(2)} = \alpha, \quad u_2^{(3)} = \alpha, \quad u_2^{(3)} = \alpha, \ldots$$

which is not optimal if $\alpha \neq 0$.

As in Example 1, a dynamical example can now be constructed. A different one is the following. Consider the system

$$\frac{dx}{dt} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u_1 + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u_2, \quad x_1(0) = -11, \quad x_2(0) = 0.$$

Both players want to steer the system to $x_1 = x_2 = 0$ as quickly as possible subject to $u_i(t) < 1, i = 1, 2$. The numerical method described gives, if we start with $u_1^{(0)} = 0, u_2^{(0)} = 0, u_1^{(1)} = 1, \ldots$
The optimal solution, however is

\[
\begin{cases}
u_1 = 1, & 0 < t < -1 + \frac{\sqrt{21}}{2}, \\ -1 < t < 1 + \frac{\sqrt{21}}{2}, \\ -1, & 1 + \frac{\sqrt{21}}{2} < t < 21,
\end{cases}
\]

\[
\eta = \sqrt{21} \approx 4.58.
\]

Had we started the numerical procedure with \(u^{(0)} = 0\), the sequence \((u^{(0)}, u^{(1)}, u^{(2)}, \ldots)\) would again converge to a wrong solution; in this case we get \(\eta = (33 - 2\sqrt{11} + 2\sqrt{11} - 1) \approx 4.66\).

**Main Result**

Let \(A\) be an \(n \times n\) matrix with eigenvalues \(\lambda_j\) satisfying (5) and let \(f(\lambda)\) be an analytic function of \(\lambda\) in a region containing all \(\lambda_j, i = 1, 2, \ldots, m\), \(j = 1, 2, \ldots, K_r\). Then,

\[
f(A) = \sum_{i=1}^{n} a_i A^{i-1}
\]

where \(a_i\) is the \(i\)th component of

\[
a = (V^{-1})^T F,
\]

\(F\) is an \(n\) vector with

\[
F_j = f^{(j-1)}(\mu), \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, K_r
\]

and \(V^{-1}\) is the inverse of the confluent Vandermonde matrix defined in (1)-(4). \(V^{-1}\) has the closed form given below (adapted to our notation from [5]):

\[
V^{-1} = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_m \end{bmatrix}
\]

where

\[
W_j^{(k)} = \frac{h_j^{(k-1)}(\mu)\mu_j^{(k-1)}}{(K_j - j - 1)!}, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, K_r
\]

Using the Cayley–Hamilton theorem to write \(A^k\) in the form

\[
A^k = \sum_{i=1}^{n} \beta_{ki} A^{i-1},
\]

we then have

\[
f(A) = \sum_{i=1}^{n} a_i A^{i-1}
\]

Note that \(f^{(k)}(\mu)\) denotes the \(k\)th derivatives of \(f(\lambda)\) with respect to \(\lambda\) evaluated at \(\lambda = \mu\).