Consider the approximate discrete-time optimization problem with dynamics:

Minimize the performance index

with

The performance index is written as such to allow the problem to be phrased as an unconstrained minimization problem, which has numerical convenience. is the interval change in parallax angle relative to feature from time to , and is defined by

Where are inertial coordinates of an agent at time . The Hamiltonian is

The usual stationarity condition is

In general, this admits only a numerical solution for . The costate equation is

The system states are inertial 2D coordinates and the controls are inertial velocity components . Here we consider both the fixed-final state and free-final state problems.

### Fixed final state

When the final state is fixed, a numerical solution is found by generating an initial and integrating backwards from the final state. The problem then becomes: determine that satisfies the governing equations such that the resulting trajectory minimizes

### Final state free

When the final state is free, we obtain the boundary condition . The problem is now to find which results in a trajectory that minimizes

Some additional steps should be noted. To solve this problem using forward integration, and must be written in terms of and . The stationarity condition must be rewritten as

and solved for , which is then used to compute and .

### Comments

The problem is framed without explicit constraints. The motivation is to enable the use of relatively simple function minimization and root-finding algorithms in the solution. The problem can be solved iteratively beginning with coarse discretizations .

The parallax term in the performance index is troublesome because is a function of both and . The Euler discretization of the governing dynamics allows us to write as a function of either or and as appropriate for forward or backward integration. However, the resulting solution for the optimal control is highly nonlinear and may not be easy to solve for large numbers of features. It is also not clear if it is possible to incorporate restrictions such as limited sensor fields of view without introducing mixed-integer type programming problems, which will complicate the solution.

It may be more appropriate to use a cost function based on the final-state covariance expectation or the Kramer-Rao lower bound.

It is not clear how/if such a planning algorithm should be incorporated into a multiagent problem. A totally decentralized solution will not gain anything, but a fully centralized solution may not be practical. Perhaps game-theoretic ideas might be used to predict the trajectories of other agents and incorporate them into a decentralized optimization scheme.