Lecture Notes in Comp. Sci. Vol. 253,
'Parallel Processing: Logic, Organization,
and Technology',pp. 29-39 (Springer,1987)

PLANNING A DYNAMIC TRAJECTORY VIA PATH FINDING IN DISCRETIZED PHASE SPACE

H. Ritter and K. Schulten
Physik-Department
Technische Universität München, D-8046 Garching

ABSTRACT

The control of dynamical systems under constraints on the the controlling force can require long term trajectory preplanning to avoid entering dead ends in phase space. To handle such problems in a flexible way we suggest to formulate them as a path finding problem on a suitably discretized version of the phase space of the system. This path finding problem can then be solved with a standard relaxation method in a parallel computation by exploiting a physical analogy: the target point in phase space is considered as the source of a diffusing substance whose concentration gradient serves as a local guide to the destination point.

I. INTRODUCTION

Many control tasks require to bring a given mechanical system in a state A, e.g. a robot arm, to a destination state B by applying a sequence of suitable control actions. Often the dynamics of the system given the pertinent restrictions on the available control force forbids a great many of the geometrically possible trajectories and leaves available only few possible paths. These paths often assume such an indirect route from A to B that search algorithms are required for their identification. An illustration of such a situation is provided by a pendulum with initial and final states A and B as shown in Fig.1. The state B shall be reached by applying a suitable torque $F(t) \in [F_{min}, F_{max}]$, where F_{min} and F_{max} are the limits on the control torque. If the torque interval $[F_{min}, F_{max}]$ is very narrow the pendulum cannot be moved directly into the

upward position B (see Fig.1), but can be moved to this state only by passing first through a few oscillatory cycles of increasing amplitude and energy about its position of stable equilibrium.

The reason for this indirect route from A to B is that from any given point x in phase space the neighboring points are not equally accessible if only small external forces are available. Actually, in the absence of external control forces, i.e., for $F_{min} = F_{max} = 0$, solely the points lying on the forward part of the phase space trajectory passing through x are accessible. For a small, but nonvanishing range of the control force the set of possible trajectories through x fills a conical region with apex at x. This region is accessible from x , all other points are blocked. The dynamics can be considered, therefore, as an obstacle blocking the access to most of the surround of x , leaving only a small subset of available paths emanating from x. This is mainly due to the inertia of mechanical systems which for finite forces restricts the range of possible accelerations.

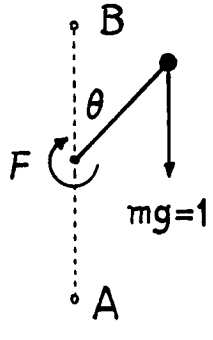

Figure 1.

We represent this situation by employing a discretization of the phase space. The continuous phase space is replaced by a discrete lattice of points x, and all lattice points in the neighborhood of x, which are accessible from x, are connected to x through directed links. The links depend on the size of the interval $[F_{min}, F_{max}]$. The phase space is then covered by a directed graph G with the lattice points as nodes, whose link structure represents the constraints imposed by the dynamics of the system and the restricted range $[F_{min}, F_{max}]$ of available control forces. The initial and final states A and B correspond to nodes x_A and x_B of G and the task of finding an admissible trajectory reduces to finding a path on G which joins x_A and x_B.

For the purpose to find such a path, we compute an auxiliary function u(x) which represents the concentration of a fictitious substance emitted at the destination point x_B and undergoing a diffusion-like spread on G. u(x) can only spread along the links of G, but opposite to their direction. Therefore starting at x_A and always following the link along which the increase of u is greatest, we obtain a path leading from x_A to x_B. Among all possible alternative paths this path is most favourable in the sense that it would be preferred by diffusion along the links.

The function u(x) can be calculated in a fully parallel manner by means of a standard relaxation algorithm ([1]). As long as the destination point x_B remains fixed u(x) can be used without any recomputation to find paths from arbitrary starting points x. This is of practical significance for if on execution of the originally intended trajectory an unprecedented disturbance shifts the current state to a new location , so that the remaining part of the intended trajectory does no longer apply, u(x) can be reemployed for the determination of the remaining path to x_B. Our present discretization scheme restricts our method to a two dimensional phase space, i.e. to systems with one degree of freedom. Improved discretization schemes requiring only very few lattice points along each dimension should render also systems with more degrees of freedom tractable.

In the subsequent Section II we describe the discretization of the phase space leading to the

graph G. In Section III the path finding method is discussed and in Section IV we illustrate the application of the method to the example mentioned above, the motion of a pendulum.

II. Phase Space Discretization

We consider a system described by an equation of motion of the form

$$\frac{d^2q}{dt^2} = R(q,\dot{q}) + F(t)\cdot M(q,\dot{q}) \tag{1}$$

where $F(t)$ is the control parameter, subsequently called "force" and R, M are given continuous functions with M nonvanishing. By $x = (q,\dot{q})$ we denote the phase space coordinates. For the purpose of the numerical computation the continuous set of 2-dimensional vectors x is represented by a lattice of $n_q \times n_{\dot{q}}$ points $(q_{ij},\dot{q}_{ij}), i = 1..n_q,\ j = 1..n_{\dot{q}}$. The lattice coordinates q_{ij} and $\dot{q}_{ij}$ are chosen as follows:

$$q_{ij} = \begin{cases} q_{min} + (i-1)\cdot\Delta q & \text{if j is odd,} \\ q_{min} + (i-1)\cdot\Delta q + \frac{1}{2}\Delta q & \text{if j is even.} \end{cases} \tag{2}$$

$$\dot{q}_{ij} = \dot{q}_{min} + (j-1)\cdot\Delta\dot{q} \tag{3}$$

The assumed lattice spacings for q and $\dot{q}$ are $\Delta q = \frac{q_{max}-q_{min}}{n_q-1}$ and $\Delta\dot{q} = \frac{\dot{q}_{max}-\dot{q}_{min}}{n_{\dot{q}}-1}$, respectively, where q_{min}, q_{max} and $\dot{q}_{min}, \dot{q}_{max}$ are the limits of q and $\dot{q}$.

The shift of q_{ij} for any other row j by $\frac{1}{2}\Delta q$ is introduced in Eq.(2) in order to have nearest neighbors in 6 directions for each point instead of in only four directions as in the case of a simple square lattice. As this number determines the degree of the directional discretization associated with replacing the continuous trajectories by chains of links connecting neighboring lattice points, we additionally include for each point the six next nearest sites into its neigborhood, thereby obtaining a total of 12 potential directions for the formation of links emanating from a given lattice point.

After the discretization of the phase space is specified we determine for each lattice point x those of its 12 lattice neighbors which are "accessible" from x and join each of them to x by a directed link emanating from x. A neighbor y is called "accessible from x" if it is possible to find a value F of the control force such that the following conditions are satisfied:

1. $F \in [F_{min}, F_{max}]$,
2. y can be reached in a time $\tau \leq \Delta t$ from x, using the linearized equation of motion with force F =const. over time τ; here Δt is a parameter which should be chosen such that the linearization of the equations of motion is a reasonable approximation over times up to Δt and that the trajectories can bridge one lattice spacing in a time significantly less than Δt.

Given $x = (q_x, \dot{q}_x), y = (q_y, \dot{q}_y)$, the values for τ and F are calculated from

$$q_y = q_x + \frac{\tau}{2}(\dot{q}_x + \dot{q}_y) \tag{4}$$

$$\dot{q}_y = \dot{q}_x + \frac{\tau}{2}\Big(R(q_x, \dot{q}_x) + R(q_y, \dot{q}_y)\Big) + \frac{1}{2}\tau \cdot F \cdot \Big(M(q_x, \dot{q}_x) + M(q_y, \dot{q}_y)\Big) \tag{5}$$

If $q_x = q_y$ and $\dot{q}_x + \dot{q}_y = 0$, Eq.(4) is valid for arbitrary τ. In this special case we choose F to minimize a positive solution for τ in (5), which, if possible, yields either $F = F_{min}$ or $F = F_{max}$.

The result of forming for each lattice point links to all its accessible neighbors yields a directed graph G, which represents a discretization of the phase space together with the possible trajectories on it. The connectivity of this graph increases with increasing interval $[F_{min}, F_{max}]$. This property reflects the fact that a wider range for the admissible control force makes more trajectories realizable. On the other hand, if the interval for the control force is too small or the discretization too coarse, the above discretization scheme may not be able to represent all possible trajectories.

The main advantage of our scheme is its ability to deal flexibly with very general constraints (e.g. position dependence) on the available control forces and the admissible system configurations. These constraints are represented solely in the graph structure of G and, once this has been done, the actual part of the algorithm operating upon G and described below runs without further consideration of the dynamics and the control forces.

III. The Path Finding Algorithm

Let A and B denote the initial and final states of the desired trajectory and x_A, x_B the two lattice points of G matching A and B closest. We search for a sequence $(x_i)_{i=1..n}$, $x_0 = x_A, x_n = x_B$ of points of G forming a linked chain connecting x_A and x_B. Once we have found such a chain we know by construction of G that the x_i can serve as successive states of the desired trajectory, as each x_i can be reached from its predecessor x_{i-1} by applying a constant force $F \in [F_{min}, F_{max}]$ over a time $\tau \leq \Delta t$.

To find such a chain we search for a real valued auxiliary function $u(x)$ defined on the nodes x obeying the following conditions:

$$u(x_B) = 1 \tag{6}$$

$$u(x) = \frac{m}{N_x} \sum_{y \in U_x} u(y) \qquad \text{if } x \neq x_B \tag{7}$$

Here U_x denotes the set of all nodes which are neighbors of x, i.e. $y \in U_x$ if and only if there is a link of G which points from x to y. N_x is the number of nodes in U_x.

The above conditions can be interpreted as follows: The destination point x_B is the source of a ficticious substance, whose concentration at node x is $u(x)$. The substance can spread only

along the links of G. A link carries a flux equal to the concentration difference between the two nodes at its ends, provided that its direction is towards the node with the higher concentration. Otherwise the flux is set to zero, i.e. the links act as "one way" connections.

At node x_B the substance is injected at such a rate as to maintain $u(x_B) = 1$ (condition (6)). In the stationary state and without any losses, condition (7) with $m = 1$ holds. Any smaller value $m \in [0,1]$ corresponds to absorptive losses at the nodes, which can be used to avoid the trivial stationary solution $u(x) = u(x_B) =$ constant for finite G.

The path leading from x_A to x_B can then be found by starting at x_A and taking as the next station always the neighbor with maximal $u(x)$, i.e the path starts at $x_0 := x_A$ and continues according to the rule $(i = 0,1,2...)$ $x_{i+1} := y$ where $u(y) = \max_{x \in U_{x_i}} u(x)$ until finally $x_n = x_B$. Since G is finite, the algorithm is guaranteed to terminate, yielding a sequence of stations $x_A = x_0, x_1, ...x_{n-1}, x_B$ which by construction constitutes a proper path. Moreover, this path is short in the sense that it takes that route towards x_B which maximizes the increase of $u(x)$ at each step.

To calculate u any procedure to solve a stationary discretized diffusion equation can be used. So far we did not try to optimize this part of the calculation and used the rather unsophisticated relaxation scheme

$$u_{i+1}(x) = \begin{cases} \frac{1}{N_x+1} \sum_{y \in U_x} u_i(y) & \text{if} \quad x \neq x_B \\ 1 & \text{else.} \end{cases} \tag{8}$$

which corresponds to the choice $m = \frac{N_x}{N_x+1}$. This scheme could be improved, e.g. by using an overrelaxation method. It is actually not strictly necessary to iterate until good convergence to the stationary solution has been obtained. Instead choosing u initially equal to zero (except at x_B) the calculation may be stopped as soon as $u_i(x_A)$ gets positive. This corresponds to "freezing" the propagation of the aforementioned substance as soon as it has reached x_A. However this may render a recomputation of u necessary if later paths from other initial points to x_B are sought.

IV. Simulation Results

We applied the above method to the task of planning a trajectory for a pendulum in a vertical gravitational field of unit acceleration. The motion of the pendulum can be affected by a torque F(t) acting at its pivot. The equation of motion is given by

$$\frac{d^2\theta}{dt^2} - \sin\theta = F(t) \tag{9}$$

The equilibrium position for this pendulum is at $\theta = 180^o$. The control task is to bring the pendulum into the inverted state $\theta = 0^o$ and stabilize it there, starting from different given initial conditions and with different admissible intervals for F(t). The discretized region of phase space is the rectangle $[-5.37, 2.49] \times [-2, 2]$ (the numbers are adjusted to yield discretization points at $\theta = -\pi$ and $\theta = 0$) in $(\theta, \dot{\theta})$-space. This phase space was represented by a mesh of 31x19 discrete nodes.

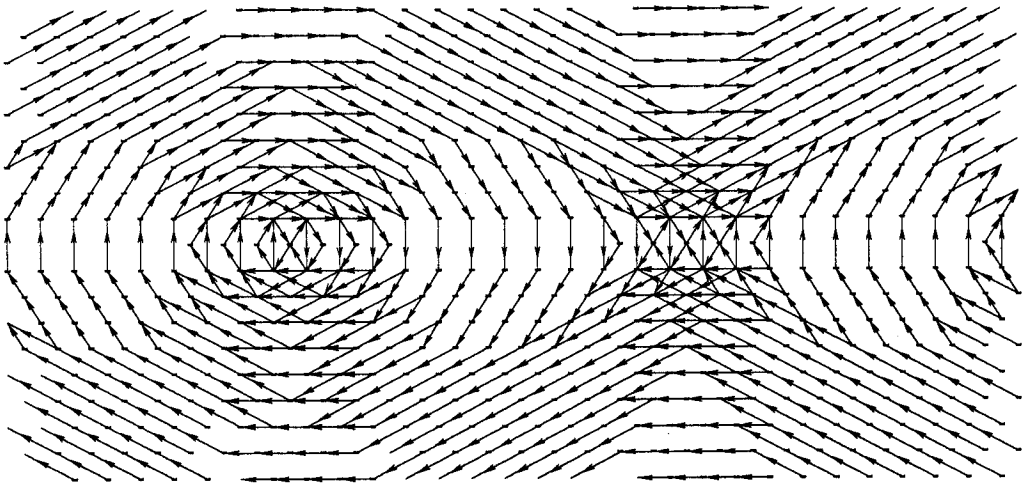

Figure 2: The phase space discretization graph G for $F \in [-0.5, 0.5]$.

Figure 2 shows the graph G, which is obtained if F(t) is restricted to the interval $[-0.5, 0.5]$. If the pendulum is initially at rest in its stable equilibrium position $x_A = (-\pi, 0)$ a direct movement into the inverted position x_B requires a torque larger than 0.5 and such movement is therefore not possible. This manifests itself in the fact that the only paths on G which connect x_A and x_B are spiralling outwards from x_A before they approach x_B. This behaviour indicates the necessity of a few oscillatory cycles of the pendulum before attaining the state x_B. Such a path, which has been obtained using the algorithm of Section II, is shown superimposed on G in Figure 3.

Figure 4 shows the trajectory which corresponds to the path of figure 3. A naive way to construct a trajectory for a path of lattice points $x_A = x_1, x_2, ...x_B$ would be to store the time steps τ_i and the force values F_i connected with each of the links $x_i \rightarrow x_{i+1}$, and then to apply these forces F_i over timesteps τ_i in an exact integration of the equation of motion, Eq.(9). However, because of the linear approximation in constructing the links in Figure 3 the deviation of the resulting trajectory from the path $x_A = x_1, x_2, ...x_B$ tends to increase in time. Therefore we have adopted an alternative procedure which keeps the actual trajectory close to the discretized path. Let x be a point reached after a certain number of integration steps and let x_j be the lattice point closest to x. We then try to reach the lattice point $y \in U_{x_j}$ for which $u(y) = \max_{z \in U_{x_j}} u(z)$.

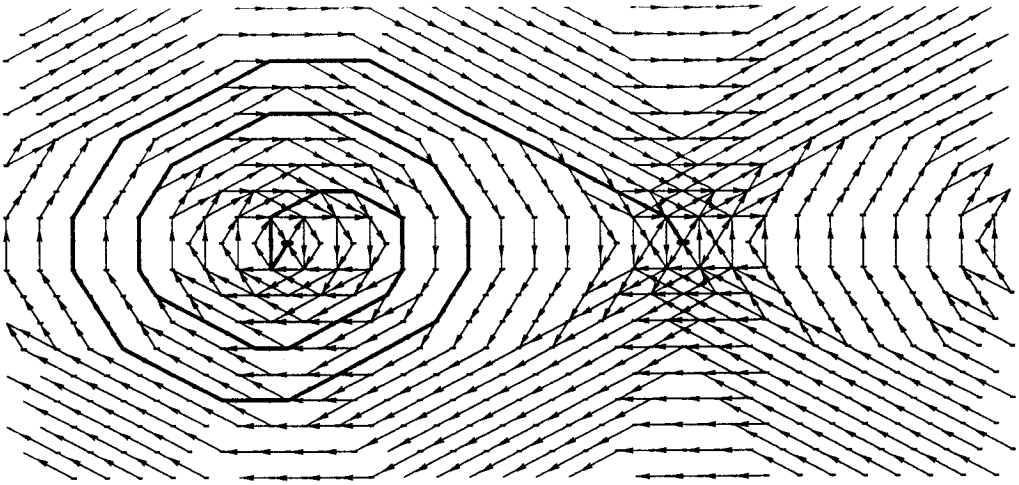

Figure 3: Trajectory starting at $x_A = (-\pi, 0)$ (stable equilibrium position of pendulum) and ending at $x_B = (0, 0)$ (unstable equilibrium position of pendulum) superimposed on the discretized phase space G.

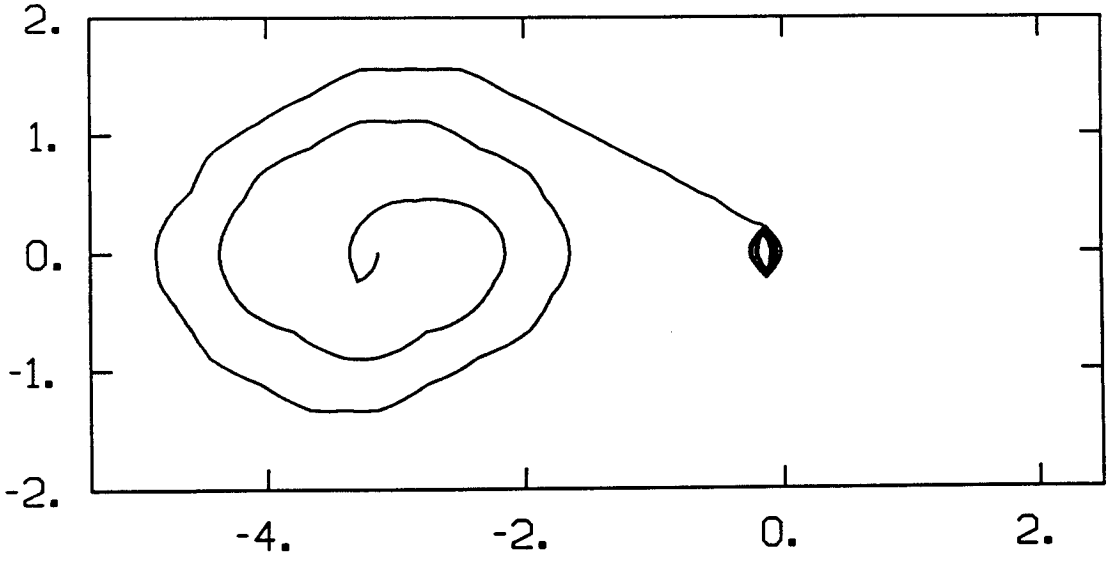

Figure 4: Simulation of the trajectory in Fig.3. The simulation starts with state $x = x_A$ at time $t = 0$ and uses equations (4) and (5) to determine the force F and the time τ required, so that the linearized trajectory passes through the neighbor y of x with $u(y)$ =maximal, i.e. $y = x_2$. These data are used to simulate the exact motion until time $t + \tau$ after which x is replaced by the new state obtained and the whole cycle is repeated. Once the unstable equilibrium position x_B has been reached, the motion gets trapped there, since any deviation from x_B means just a new starting value x_A which, however, is already very close to x_B.

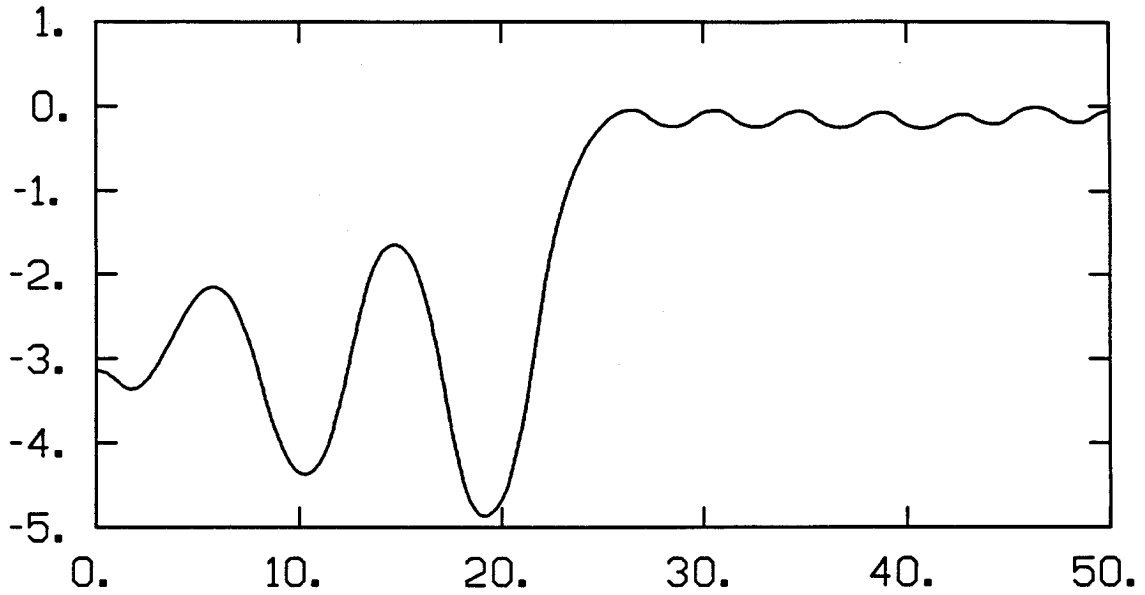

Figure 5: Angle $\theta(t)$ for $t \in [0, 50]$ for the trajectory in Figure 4.

The corresponding force F and time step τ are determined according to Eqs. (4) and (5). These quantities are then applied for the next integration step. The described procedure is then repeated. In this way the simulated trajectory follows the path of Figure 3 until x_B is reached.

As x_B is no stable equilibrium point, the system cannot stay there. The best it can do is to choose as next station that neighbor y of x_B with $u(y) = \max_{z \in U_{x_B}} u(z)$, although now this value is less than $u(x_B)$. In our example this turns out to be the lattice point "south–west" of x_B. But after this one inescapable step away from x_B to y, the further steps again are directed back to x_B, because y is just a new initial state like formerly x_A. This yields the small "diamond–shaped" limit cycle visible in Figure 4, which contains x_B as its "east corner". The effect can be seen still more clearly in Figure 5, which depicts the angle $\theta(t)$ after the system has been released from x_A. First $\theta(t)$ undergoes a few cycles of increasing amplitude until it reaches its final value $\theta_B = 0°$ and remains there with the small limit cycle oscillations superimposed.

Figure 6 shows the constructed path and Figure 7 the simulated trajectory for a different initial position x_A. In this second example again no direct path is possible from x_A to x_B but rather the pendulum first has to be swung back and then forth before x_B can be reached.

If the admissible intervall for the torque F(t) is larger the connectivity of G increases and additional paths open up. This is shown in Figure 8, which compares the graphs G for the three F–intervals $[-0.5, 0.5]$, $[-0.75, 0.75]$ and $[-1, 1]$, corresponding to weak, medium and strong control torques, respectively. The corresponding trajectories from the initial downward resting state x_A to the inverted state x_B are superimposed for each case. This illustrates the effect of the range of the available control torque on the shape of the trajectories obtained. For weak torques the pendulum

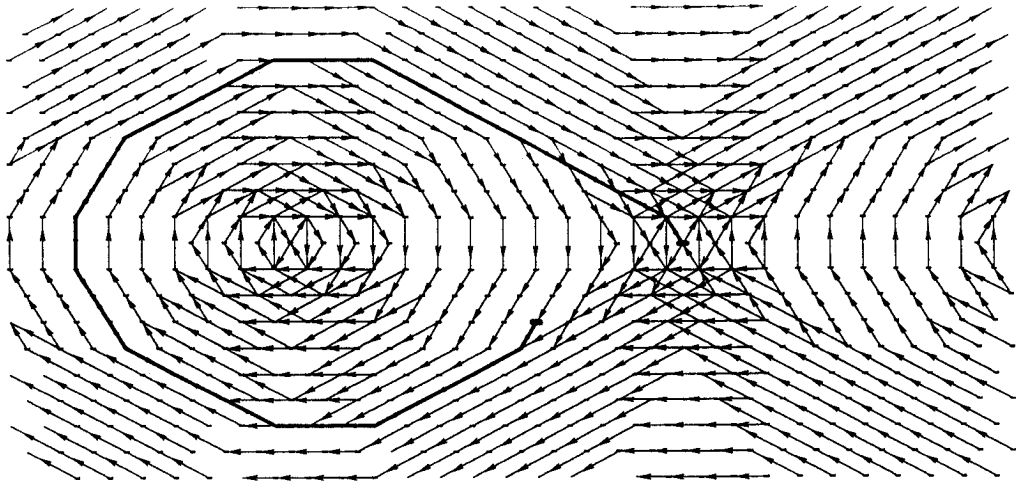

Figure 6: Trajectory starting from a different initial value $x_A = (-1.17, -0.67)$ and ending at $x_B = (0, 0)$ superimposed on discretized phase space G.

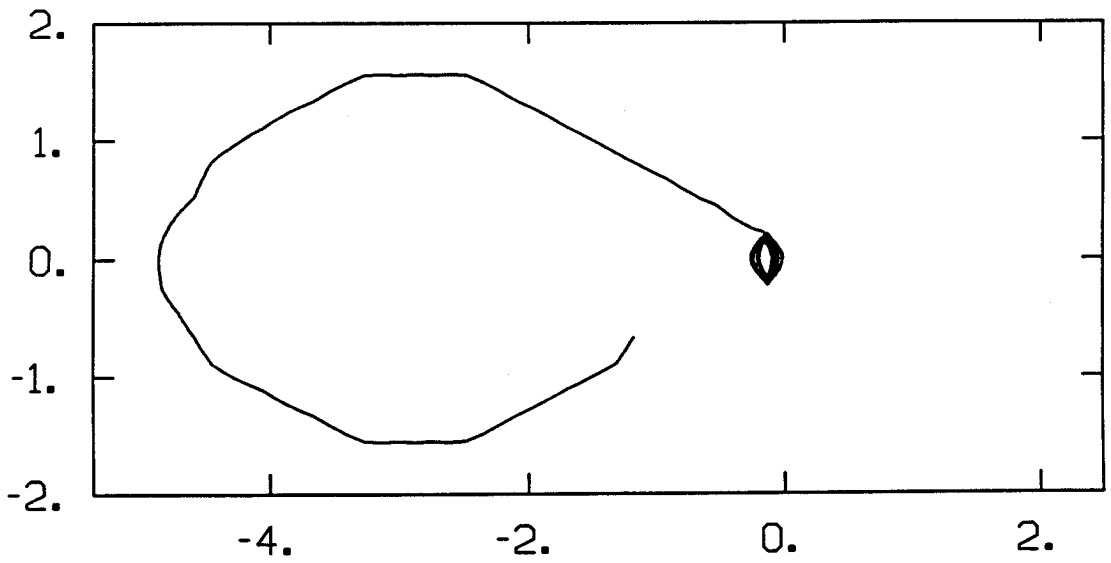

Figure 7: Simulation of the trajectory in Figure 6 as explained in the text.

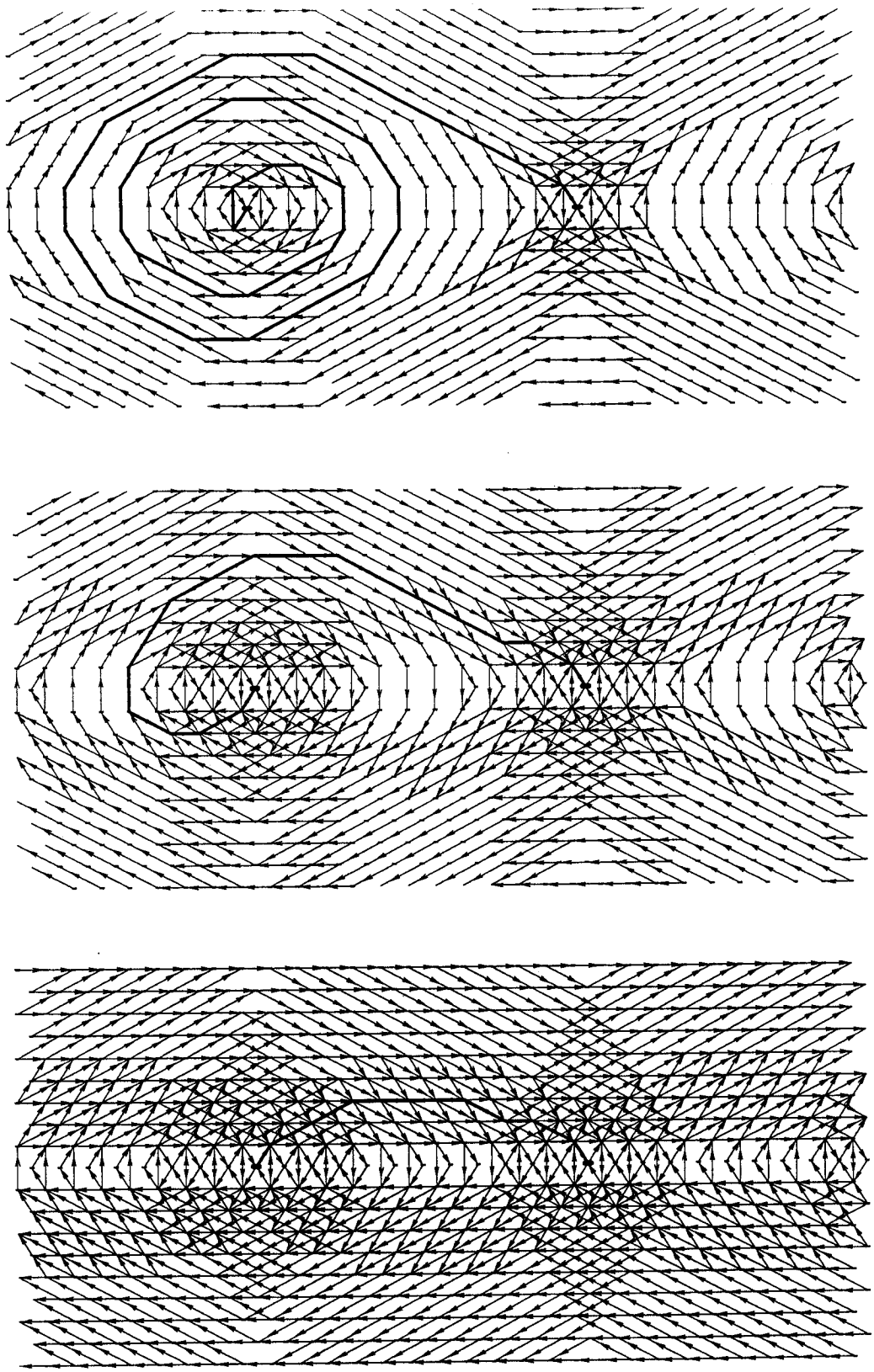

Figure 8: Phase space discretization graph G with trajectory from $x_A = (-\pi, 0)$ to $x_B = (0, 0)$ for $F \in [-0.5, 0.5]$ *(top)*, $F \in [-0.75, 0.75]$ *(middle)* and $F \in [-1, 1]$ *(bottom)*.

has to be swung back and forth several times before it can be moved to x_B *(top)*, whereas for medium torques one previous cycle suffices *(middle)*. Strong torques allow turning the pendulum directly from x_A to x_B *(bottom)*.

V. Conclusion

We have presented a method for control tasks of mechanical systems. The method treats the task as a path finding problem in the phase space of the system. Using a physical analogy we can solve this problem by means of a relaxation method together with a suitable discretization scheme. This approach enables us to handle constraints on the admissible control forces and system configurations in a very flexible manner. However the discretization scheme employed here limits the method to systems with one degree of freedom only. We are currently extending the method to the control of higher dimensional mechanical systems.

References

[1] William H. Press, Brian P. Flannery, Saul A. Teukolsky, William T. Vetterling: Numerical Recipes, Cambridge (1986).