Pole-Placement & Machine-Learning Control for N-level Open Quantum Systems

Alexei Stepanenko

January 9, 2018

1 Introduction

The state-space for a N-level (open) quantum system is a unit ball in $\mathbb{R}^d$ where $d = (N+1)(N-1)$. In this document we derive a simple controller for a such a system using elementary techniques from control theory. This technique, called pole-placement works by manipulating the eigen-values of a fixed point or periodic orbit. In order in account for non-linearities in the noise and non-uniformity of the control vector-field we outline a method based on machine learning (on-line minimisation of a cost function).

2 Linearisation

The stochastic master equation can be written in co-ordinate form (and Stratonivich form),

$$\dot{\mathbf{q}} = A_0\mathbf{q} + A_u\mathbf{q}u + g_L(\mathbf{q})\xi. \tag{1}$$

(1) is non-linear in two ways. Firstly, with a linear controller the term $A_u\mathbf{q}u$ becomes quadratic. To linearise this term we must make it constant. Let $\mathbf{q}_d(t) = e^{A_0(t-t_0)}\mathbf{q}_0$ be the desired trajectory. Now we use a 'self-consistent' arguement. Assume that we can apply an effective linear control scheme. Then, $|\mathbf{q} - \mathbf{q}_d|$ is small, hence, we can approximate $A_u\mathbf{q}$ by $A_u\mathbf{q_d}$.

The second non-linear term is the diffusion term $g_L(\mathbf{q})$. In this document, we treat the noise (whose origin in weak measurement) as unwanted disturbances and simply neglect it. Note that this is different from averaging ove the noise as we are not left with a damping term (a Linblad superoperator term in co-ordinate form).

The linearised (and de-noisified) equations read,

$$\dot{\mathbf{q}} = A_0\mathbf{q} + A_u\mathbf{q}_d u. \tag{2}$$

3 Linear Pole-Placement controller

Let $\mathbf{r} := e^{-A_0 t}\mathbf{q}$. Then,

$$\begin{aligned}\dot{\mathbf{r}} &= e^{-A_0 t} A_u e^{A_0 t}\mathbf{r}u \\ &\approx e^{-A_0 t} A_u e^{A_0 t}\mathbf{q}_0 u. \end{aligned} \tag{3}$$

Let $A_u^t := e^{-A_0 t} A_u e^{A_0 t}$. Let the controller have a linear form, $u = -\mathbf{k}^T\boldsymbol{\Delta}$ where $\boldsymbol{\Delta} := \mathbf{r} - \mathbf{q}_0$ and the control vector $\mathbf{k}$ is to be determined . Then, (3) has form,

$$\begin{aligned}\dot{\mathbf{r}} &= -A_u^t \mathbf{q}_0 \mathbf{k}^T \boldsymbol{\Delta} \\ \Rightarrow \dot{\boldsymbol{\Delta}} &= -A\boldsymbol{\Delta} \end{aligned} \tag{4}$$

where $A := A_u^t \mathbf{q}_0 \mathbf{k}^T$.

The aim of the control is to manipulate the trajectory to fall onto the desired trajectory as quickly as possible. To achieve this we need to choose the control vector $\mathbf{k}$ so that A has eigenvalues with positive real parts with the positive parts having as high magnitude as possible. **There are many different choices for the eigenvalues**. We choose the simplest one.

Let A have fully degenerate real, positive eigenvalues with magnitude λ. Then,

$$A = \lambda I_d$$

where I_d is the d-dimensional identity matrix. Then, letting $\mathbf{b} := \frac{1}{\lambda} A_u^t \mathbf{q}_0$,

$$\mathbf{b}\mathbf{k}^T = I. \tag{5}$$

Let $\mathbf{a}$ be such that for any $\mathbf{k}$,

$$\mathbf{a}^T \mathbf{b}\mathbf{k}^T = \mathbf{k}^T. \tag{6}$$

Then,

$$\begin{aligned}\mathbf{a} \cdot \mathbf{b} &= 1 \\ \Rightarrow |\mathbf{a}| \cos\theta &= \frac{1}{|\mathbf{b}|}. \end{aligned} \tag{7}$$

Choose $\theta = 0$ so that $\mathbf{a}$ points in the same direction of $\mathbf{b}$. Then, by (7) it follows that,

$$\mathbf{a} = \frac{\mathbf{b}}{|\mathbf{b}|^2}.$$

By (5) and (6), $\mathbf{k} = \mathbf{a}$, so,

$$\mathbf{k} = \lambda \frac{A_u^t \mathbf{q}_0}{|A_u^t \mathbf{q}_0|^2}.$$

The controller follows,

$$u = -\lambda \frac{A_u^t \mathbf{q}_0}{|A_u^t \mathbf{q}_0|^2} \cdot (e^{-A_0 t}\mathbf{q} - \mathbf{q}_0) \tag{8}$$

The free parameter λ in (8) defines the overall strength of feedback. The factor $\frac{1}{|A_u^t \mathbf{q}_0|}$ in (8) modulates the feedback strength according to the strength of the control vector-field to manipulate the trajectory at the desired trajectory at time t. If the control vectorfield is zero at any point on the desired trajectory this factor can become singular. In this case useful to set $\frac{1}{|A_u^t \mathbf{q}_0|}$=1. This would also be a desirable simplication if the capacity of the control vector-field stays approximatly constant along the trajectory. A simplified controller reads,

$$u = -\lambda A_u^t \mathbf{q}_0 \cdot (e^{-A_0 t}\mathbf{q} - \mathbf{q}_0). \tag{9}$$

4 Machine-Learning Control

The controllers (8) & (9) were derived using many simplications & choices and are certainly not optimal. To obtain a more optimal controller, we first write down a cost functional $J(\{\mathbf{q}, u\}_t)$ which we wish to minimise.

Let $\mathbf{q} = (p, \boldsymbol{\theta})$ be the generalised Bloch vector in generalised spherical co-ordinates. The control vector-field is defined by a Hamiltonian, $A_u \mathbf{q} = \iota(-i[H_u, \rho])$, - it spins the vector about an axis. It therefore has no effect on the length of the vector (the purity), hence, the controller should not depend on p. Hence we write,

$$u = -\boldsymbol{\kappa} \cdot \boldsymbol{\theta}$$

for some time dependant vector $\boldsymbol{\kappa}$.

Furthmore, the time dependance of the controller should be periodic with the period of the desired trajectory. We can therefore write the the time evolution of the components of $\boldsymbol{\kappa}$ as a Fourier series . Let Ω denote the frequency of the desired trajectory. Then, the most general form for the controller is,

$$u = -\sum_{j=0}^{d-1} \sum_{n=1}^{\infty} (a_{jn} \cos(n\Omega t) + b_{jn} \sin(n\Omega t))\, \theta_j \tag{10}$$

where $\theta_0 := 1$.

Truncating the values of n in (10), we can write all the components a_{jn} and b_{jn} defining the controller in terms of a large vector $\mathbf{P}$. We define the minimisation function $M(\mathbf{P})$ as the expected value of the cost functional $J(\{\mathbf{q}, u\}_t)$ given that the controller is defined by the the parameters $\mathbf{P}$. The control scheme is then summarised as the following:

1. Find $\mathbf{P}$ so that the controller mimics linear controllers (8) or (**??**).
2. Find $\mathbf{P}$ so that $M(\mathbf{P})$ is minimised using stochastic gradient descent algorithms