1-2021

# Boolean Network Control with Ideals

Ian H. Dinwoodie
*Portland State University*, ihd@pdx.edu



## Citation Details

Dinwoodie, Ian H., "Boolean Network Control with Ideals" (2021). *Mathematics and Statistics Faculty Publications and Presentations*. 302.
https://pdxscholar.library.pdx.edu/mth_fac/302

# Boolean Network Control with Ideals

I H Dinwoodie
Portland State University

January 2021

**Abstract**

A method is given for finding controls to transition an initial state $\mathbf{x}_0$ to a target set in deterministic or stochastic Boolean network control models. The algorithms use multivariate polynomial algebra. Examples illustrate the application.

## 1 Introduction

A Boolean network is a dynamical system on $d$ nodes (or coordinates) with binary node values, and with transition map $F : \{0,1\}^d \to \{0,1\}^d$. The coordinate maps $F = (f_1, \ldots, f_d)$ may use binary parameters $\mathbf{u} \in \{0,1\}^c$ that can be adjusted or controlled at each step, so the image can be written $F(\mathbf{x}, \mathbf{u})$ and $F$ takes $\{0,1\}^{d+c} \to \{0,1\}^d$, usually written in logical notation. Then a standard problem is to find ways to make a given initial point $\mathbf{x}_0$ move to a target set $A \subset \{0,1\}^d$ in $M$ steps with a sequence of controls $\mathbf{u}_0(\mathbf{x}_0), \mathbf{u}_1(\mathbf{x}_1), \ldots, \mathbf{u}_{M-1}(\mathbf{x}_{M-1})$. That is, we seek to achieve:

$$
\begin{aligned}
F(\mathbf{x}_0, \mathbf{u}_0) &= \mathbf{x}_1 \\
F(\mathbf{x}_1, \mathbf{u}_1) &= \mathbf{x}_2 \\
&\cdots \\
F(\mathbf{x}_{M-1}, \mathbf{u}_{M-1}) &= \mathbf{x}_M \in A
\end{aligned}
$$

which is called an M-step control.

It is of interest to consider stochastic dynamics as well [15], where the map $F$ is chosen randomly from a collection of transformations $F^w : \{0,1\}^{d+c} \to \{0,1\}^d, w = 1, 2, \ldots, \tau$ and this extension is treated. The index $w$, flagging which transformation, is selected independently at each time step for the methods here.

Methods below for the existence and construction of controls are based on multivariate polynomials, or commutative algebra. A set of binary states is described as the roots of a polynomial system called an "ideal." Polynomial operations for logic operations go back to [2] and [19] and recent developments are described in, for example, [9]. The algebra involved is standard, with clear expositions in [4] and [14]. Software packages for implementing the algorithms are [1], [5] (used for calculations in the examples) and [8].

Control problems in discrete dynamics are well-established, going back to [11] and [17], so there are several approaches. Semi-tensor algebra ([3], [15]) embeds the process in a high-dimensional linear space. Dynamic programming [16] and aggregation [18] may help with large problems. The polynomial methods will also be useful as they are very clear and may handle some large problems as well.

## 2 Existence of Controls

First we describe an algorithm for determining if a control sequence of length $M$ exists for moving starting state $\mathbf{x}_0$ into set $A$. In the next section we give an algorithm for finding controls to move the starting state forward.

Suppose we have an integer $\tau$ of possible Boolean transition maps $F^w$ where the index $w$ says which map is chosen randomly (and independently at each step) from an indexing set $\{1, 2, \ldots, \tau\}$. The random selection of a sequence $w_1, w_2, \ldots, w_M$ will assume independent coordinates with each coordinate distribution giving positive probability to each value. The controls will use the present state and the current outcome of the transformation index process $(\mathbf{u}_0 = \mathbf{u}_0(\mathbf{x}_0, w_1), \mathbf{u}_1 = \mathbf{u}_1(\mathbf{x}_1, w_2), \ldots)$, a form of feedback control.

The polynomial algebra is set up with indeterminates for the $c$ controls and for both forward and backward node values, giving a total of $2d + c$ variables, similar to [7]. The algebraic operations find preimages of the transition maps in alternating sets of indeterminates. We will use finite field coefficients for speed and memory efficiency, although the complex numbers can also be used and simplify some of the theory (being algebraically closed).

Define the ring of polynomials

$$R := K[s_1, \ldots, s_d, v_1, ..., v_c, t_1, \ldots, t_d,] = K[\mathbf{s}, \mathbf{v}, \mathbf{t}]$$

with field coefficients $K = Z_2$, just binary numbers.

Intuitively, the $\mathbf{s}$ are for starting states, the $\mathbf{v}$ are for controls which may use them, and the $\mathbf{t}$ are the image variables under one of the transformations (but which of $\mathbf{s}$ and $\mathbf{t}$ is the domain and which is the image alternates). The indeterminates are not mathematically the same as arbitrary states $\mathbf{x}$ or controls $\mathbf{u}$, so the variable notation in the polynomial ring is slightly different for correctness. If $I$ is an ideal in the polynomial ring $R$, the operation $I \cap K[\mathbf{s}]$ is "elimination," which takes the subset of $I$ only involving indeterminates $\mathbf{s}$ (elimination issues related to finite field coefficients are settled in [6]). This operation is the hard or "complex" one in the method.

Let $A$ be a target set of interest, possibly but not necessarily an attractor. In applications this will be a collection of desirable outcomes of a biological model, such as health wellness. Realistic current examples are shown in [13], where the number of states $d$ can exceed 50. Write the initial state $\mathbf{x}_0 = (x_{0,1}, x_{0,2}, \ldots, x_{0,d})$.

Define ideals

$$
\begin{aligned}
I_{01} &= \langle s_1^2 - s_1, \ldots, s_d^2 - s_d, v_1^2 - v_1, \ldots, v_c^2 - v_c, \\
&\quad t_1^2 - t_1, \ldots, t_d^2 - t_d \rangle \\
F_{svt}^w &= \langle f_1^w(\mathbf{s}, \mathbf{v}) - t_1, f_2^w(\mathbf{s}, \mathbf{v}) - t_2, \ldots, f_d^w(\mathbf{s}, \mathbf{v}) - t_d \rangle \\
F_{tvs}^w &= \langle f_1^w(\mathbf{t}, \mathbf{v}) - s_1, f_2^w(\mathbf{t}, \mathbf{v}) - s_2, \ldots, f_d^w(\mathbf{t}, \mathbf{v}) - s_d \rangle \\
I_A &= \cap_{\mathbf{x} \in A} \langle t_1 - x_1, \ldots, t_d - x_d \rangle \\
I_{0,s} &= \langle s_1 - x_{0,1}, \ldots, s_d - x_{0,d} \rangle \\
I_{0,t} &= \langle t_1 - x_{0,1}, \ldots, t_d - x_{0,d} \rangle \\
I_0 &= I_{0,s} + I_{0,t}.
\end{aligned}
$$

Above, the polynomial $f_1^w(\mathbf{s}, \mathbf{v})$ is a polynomial representation of the first coordinate map in $F^w = (f_1^w, f_2^w, \ldots, f_d^w)$.

Define the ideal $B_0 = I_A$ and subsequent "backward" ideal $B_1^w$ for each transformation index $w$ by

$$B_1^w = (F_{svt}^w + B_0 + I_{01}) \cap K[\mathbf{s}],$$

then for the preimage of $A$ over all $w$:

$$B_1 = B_1^1 + B_1^2 + \cdots + B_1^\tau.$$

Continue recursively for a sequence of backwards ideals $B_1, B_2, B_3, \ldots, B_M$ in alternating sets $K[\mathbf{s}], K[\mathbf{t}]$ :

$$B_i^w = (F_{ts}^w + B_{i-1} + I_{01}) \cap \ K[\mathbf{t}], \ i \text{ even}, \tag{1}$$

$$B_i^w = (F_{st}^w + B_{i-1} + I_{01}) \cap \ K[\mathbf{s}], \ i \text{ odd}, \tag{2}$$

$$B_i = \sum_{w=1}^{\tau} B_i^w, \quad i = 1, 2, 3, \ldots, M. \tag{3}$$

Now we state the usefulness of the $B_i$.

**Theorem 1** *M-step controls for starting point $\mathbf{x}_0$ exist with probability 1 if and only if $B_M \subset I_0$.*

*Remark* Containment of the ideals is established simply using division by a Groebner basis for $I_0$.

**Proof** The ideal $B_1^w$ is the set of polynomials with roots equal to the set

$$\{\mathbf{x} : \exists \mathbf{u}_x : F^w(\mathbf{x}, \mathbf{u}_x) \in A\}$$

by the Extension Theorem on the elimination operation (the finite field $K = F_2$ is not algebraically closed but adding $I_{01}$ still makes it valid). Then the ideal sum $B_1 = B_1^1 + B_1^2 + \cdots + B_1^\tau$ corresponds to the intersection of sets, written as $V(B_1)$:

$$\begin{aligned} V(B_1) &= \cap_w \{\mathbf{x} : \exists \mathbf{u}_{x,w} : F^w(\mathbf{x}, \mathbf{u}_{x,w}) \in A\} \\ &= \{\mathbf{x} : \forall w \; \exists \mathbf{u}_{x,w} : F^w(\mathbf{x}, \mathbf{u}_{x,w}) \in A\}. \end{aligned}$$

Now a point $\mathbf{x}$ can map to $A$ for every possible randomly selected $w$ with some control $\mathbf{u}(\mathbf{x}, w)$ if and only if $\mathbf{x} \in V(B_1)$. Thus the set of points with a certain 1-step control is $B_1$. For the corresponding set $V(B_1)$ to contain the point $\mathbf{x}_0$ it is necessary and sufficient that $B_1 \subset I_{0,s}$, which is equivalent to $B_1 \subset I_0$. The argument repeats for $B_2, B_3, \ldots, B_M$, establishing them as controllable sets for all future transformation indices.

Now we consider trajectories forward from the initial state $\mathbf{x}_0$. First, we find possible values of $\mathbf{x}_i$, for a given sequence of transformations $w_1, w_2, \ldots, w_M$.

If $M$ is odd then the backward M-step ideal $B_M$ is in $\mathbf{s}$ variables like $B_1$. So for odd $M$ define ideal $X_0 = I_{0,s}$ for the starting point and "forward" $Xs_i$ ideals (intentionally plural) for possible states $\mathbf{x}$:

$$\begin{aligned} Xs_1 &= (X_0 + F_{svt}^{w_1} + B_{M-1} + I_{01}) \cap K[\mathbf{t}] \\ Xs_2 &= (Xs_1 + F_{tvs}^{w_2} + B_{M-2} + I_{01}) \cap K[\mathbf{s}] \\ &\cdots \\ Xs_M &= (Xs_{M-1} + F_{svt}^{w_M} + B_0 + I_{01}) \cap K[\mathbf{t}]. \end{aligned}$$

If $M$ is even, define $X_0 = I_{0,t}$

$$\begin{aligned} Xs_1 &= (X_0 + F_{tvs}^{w_1} + B_{M-1} + I_{01}) \cap K[\mathbf{s}] \\ Xs_2 &= (Xs_1 + F_{svt}^{w_2} + B_{M-2} + I_{01}) \cap K[\mathbf{t}] \\ &\cdots \\ Xs_M &= (Xs_{M-1} + F_{svt}^{w_M} + B_0 + I_{01}) \cap K[\mathbf{t}]. \end{aligned}$$

Now the use of the $Xs_i$ ideals is that each coordinate $\mathbf{x}_i$ in a possible trajectory $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_M \in A$ is a root of polynomials in $Xs_i$. Similarly, one can construct ideals for effective control values at each time step.

This result gives possible values of the coordinates at each time step. The complete trajectory with controls $\mathbf{u}_i$ is described next.

# 3 Construction of Controls

With the ideal containment $B_M \subset I_0$, it remains to find the controls that move $\mathbf{x}_0$ forward into $A$. The controls and the path may not be unique but the procedure below will identify all possibilities for randomly chosen sequences of Boolean transformations.

In the ideal operations that follow, the ideals in their respective subrings after elimination are 0-dimensional (for a finite number of roots) and radical, and each prime component corresponds to a particular solution. The notation $I[p_1]$ will denote any prime component of the prime decomposition, essentially one solution in the ideal format.

If $M$ is odd then the backward M-step ideal $B_M$ is in $\mathbf{s}$ variables like $B_1$. So for odd $M$ define ideal $X_0 = I_{0,s}$ for the starting point and *trajectory ideals* for controls $\mathbf{u}$ and states $\mathbf{x}$:

$$\begin{aligned}
UX_{(0,1)} &= (X_0 + F^{w_1}_{svt} + B_{M-1} + I_{01}) \cap K[\mathbf{v}, \mathbf{t}] \\
U_0 &= UX_{(0,1)}[p_1] \cap K[\mathbf{v}] \\
X_1 &= UX_{(0,1)}[p_1] \cap K[\mathbf{t}] \\
UX_{(1,2)} &= (X_1 + F^{w_2}_{tvs} + B_{M-2} + I_{01}) \cap K[\mathbf{s}, \mathbf{v}] \\
U_1 &= UX_{(1,2)}[p_2] \cap K[\mathbf{v}] \\
X_2 &= UX_{(1,2)}[p_2] \cap K[\mathbf{s}] \\
&\dots \\
UX_{(M-1,M)} &= (X_{M-1} + F^{w_M}_{svt} + B_0 + I_{01}) \cap K[\mathbf{v}, \mathbf{t}] \\
U_{M-1} &= UX_{(M-1,M)}[p_M] \cap K[\mathbf{v}] \\
X_M &= UX_{(M-1,M)}[p_M] \cap K[\mathbf{t}].
\end{aligned} \tag{4}$$

If $M$ is even, define $X_0 = I_{0,t}$ and trajectory ideals

$$\begin{aligned}
UX_{(0,1)} &= (X_0 + F^{w_1}_{tvs} + B_{M-1} + I_{01}) \cap K[\mathbf{s}, \mathbf{v}] \\
U_0 &= UX_{(0,1)}[p_1] \cap K[\mathbf{v}] \\
X_1 &= UX_{(0,1)}[p_1] \cap K[\mathbf{s}] \\
UX_{(1,2)} &= (X_1 + F^{w_2}_{svt} + B_{M-2} + I_{01}) \cap K[\mathbf{t}, \mathbf{v}] \\
U_1 &= UX_{(1,2)}[p_2] \cap K[\mathbf{v}] \\
X_2 &= UX_{(1,2)}[p_2] \cap K[\mathbf{t}] \\
&\dots \\
UX_{(M-1,M)} &= (X_{M-1} + F^{w_M}_{tsv} + B_0 + I_{01}) \cap K[\mathbf{t}, \mathbf{v}] \\
U_{M-1} &= UX_{(M-1,M)}[p_M] \cap K[\mathbf{v}] \\
X_M &= UX_{(M-1,M)}[p_M] \cap K[\mathbf{t}].
\end{aligned} \tag{5}$$

A *trajectory solution* to the ideals $UX_{(i-1,i)}$ is a sequence $(\mathbf{u}_0, \mathbf{x}_1), (\mathbf{u}_1, \mathbf{x}_2), \dots, (\mathbf{u}_{M-1}, \mathbf{x}_M)$ such that $(\mathbf{u}_0, \mathbf{x}_1)$ solves $UX_{(0,1)}$, $(\mathbf{u}_1, \mathbf{x}_2)$ solves $UX_{(1,2)}$, and in general $(\mathbf{u}_{i-1}, \mathbf{x}_i)$ solves $UX_{(i-1,i)}, i = 1, \dots, M$.

Note that the control $\mathbf{u}_{i-1}, i = 1, \dots, M$ depends only on the transformation indices $w_1, \dots, w_i$ and states $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{i-1}$.

**Theorem 2** *For any starting state $\mathbf{x}_0$ such that $B_M \subset I_0$, solutions to the trajectory ideals for choices of prime component indices $p_1, p_2, \dots, p_M$ give sure trajectories of controls and states $(\mathbf{u}_0, \mathbf{x}_1), \dots, (\mathbf{u}_{M-1}, \mathbf{x}_M)$ with $\mathbf{x}_M \in A$ for any sequence $w_1, w_2, \dots, w_M$ of randomly selected transformation indices.*

**Proof** The roots of the set of polynomials $UX_{(0,1)}$ in the relevant indeterminates are the $(\mathbf{u}_0, \mathbf{x}_1)$ values (which depend on $w_1$ and $\mathbf{x}_0$) that start at $\mathbf{x}_0$ and arrive in the next controllable set $B_{M-1}$ under $F^{w_1}(\mathbf{x}_0, \mathbf{u}_0) = \mathbf{x}_1$ by the Extension Theorem of algebraic geometry ([4]). Then $U_{i-1}$ and $X_i$ are the ideals for the coordinate values of a particular selection indexed by the component $p_1$. The argument moves forward for $i = 1, 2, \ldots, M$.
*Remark* To find just one trajectory without examining the prime decomposition of an ideal like (5), simply take the first component as there will always be at least one.

# 4 Examples

**Example 1.** The traditional Boolean network with values in $\{0, 1\}$ at each node is exemplified by the *lac* operon network formulated in [11] and used recently in [15]. The basic deterministic dynamics $F = (f_1, f_2, f_3)$ with three nodes are

$$\begin{aligned} f_1(x_1, x_2, x_3) &= !u_1 \ \& \ (x_3 \mid u_2) \\ f_2(x_1, x_2, x_3) &= x_1 \\ f_3(x_1, x_2, x_3) &= !u_1 \ \& \ ((x_2 \& u_2) \mid (x_3 \& !x_2)) \end{aligned}$$

where $u_1, u_2 \in \{0, 1\}$ are adjustable parameters for control that can be set at 0 or 1, "!" means "not" and "|" means "or."

A problem is to find values of $u_1, u_2$ that lead to state $\mathbf{x}_M = (1, 1, 1)$ (which means the operon is "on") so the target set $A = \{(1, 1, 1)\}$. The deterministic model sets $\tau = 1$ in the computation of $B_i$ at (3). Calculation in [5] gives $B_1 = \{s_1 - 1, (s_2 - 1)(s_3 - 1), s_2^2 - s_2, s_3^2 - s_3\}$ (with roots (1,0,1),(1,1,0),(1,1,1)) then $B_2 = \{t_1^2 - t_1, t_2^2 - t_2, t_3^2 - t_3, (t_1 - 1)(t_2 - 1)(t_3 - 1)\}$ (with seven roots not including (0,0,0)). Finally $B_3 = \{s_1^2 - s_1, s_2^2 - s_2, s_3^2 - s_3\}$ whose roots in $\mathbf{s}$ are all binary triples, including the starting state (0,0,0). The steady controls $\mathbf{u}_0 = \mathbf{u}_1 = \mathbf{u}_2 = (0, 1)$ are one possibility to reach the target state (1,1,1) with $M = 3$.
**Example 2.** Extending the deterministic method ($\tau = 1$) to stochastic ($\tau > 1$) as described in (1)-(3) can be illustrated with a randomized version of the above. We take $\tau = 2$ for two transformations at each step chosen randomly from the map $F$ of Example 1 and the map $G$ defined below (where the | is switched to & in $f_1$):

$$\begin{aligned} g_1(x_1, x_2, x_3) &= !u_1 \ \& \ (x_3 \ \& \ u_2) \\ g_2(x_1, x_2, x_3) &= x_1 \\ g_3(x_1, x_2, x_3) &= !u_1 \ \& \ ((x_2 \& u_2) \mid (x_3 \& !x_2)) \,. \end{aligned}$$

Backward ideals stabilize at $B_4 = \langle t_1^2 - t_1, t_2^2 - t_2, t_3^2 - t_3, (t_1 - 1)(t_2 - 1)(t_3 - 1)\rangle$. This means that every initial state $\mathbf{x}_0$ except the point $(0, 0, 0)$ can be controlled to hit $(1, 1, 1)$ in 4 steps, as $(0, 0, 0)$ is the only state that is not a root of the polynomial system.

We will start at $\mathbf{x}_0 = (1,0,0)$ with ideal $X_0 = I_{0,t} = \langle t_1 - 1, t_2, t_3\rangle$ to illustrate the stochastic method. We find the M-step trajectories to $(1,1,1)$ with M=4, trying for $\mathbf{x}_4 = (1,1,1)$. Suppose the transition maps are $F, G, F, G$ from a hypothetical random process of transformation indices $w_1 = 1, w_2 = 2, w_3 = 1, w_4 = 2$.

The prime decomposition of $UX_{(0,1)}$ shows four components (below is Singular output that uses the primdec.lib library) corresponding to values of $\mathbf{u}_0$ and $\mathbf{x}_1$, each of which starts a path to $(1,1,1)$ in terms of $\mathbf{x}$ and $\mathbf{u}$ values.

```
> minAssGTZ(UX01);
[1]:
   _[1]=v(2)+1
   _[2]=v(1)+1
   _[3]=s(3)
   _[4]=s(2)+1
   _[5]=s(1)+v(1)*v(2)+v(2)
[2]:
   _[1]=v(2)+1
   _[2]=v(1)
   _[3]=s(3)
   _[4]=s(2)+1
   _[5]=s(1)+v(1)*v(2)+v(2)
[3]:
   _[1]=v(2)
   _[2]=v(1)+1
   _[3]=s(3)
   _[4]=s(2)+1
   _[5]=s(1)+v(1)*v(2)+v(2)
[4]:
   _[1]=v(2)
   _[2]=v(1)
   _[3]=s(3)
   _[4]=s(2)+1
   _[5]=s(1)+v(1)*v(2)+v(2)
```

There are four trajectories in $(\mathbf{u}_{i-1}, \mathbf{x}_i)$, and two trajectories in $\mathbf{x}$ values:

$$\mathbf{x}_0 = (1,0,0), \mathbf{x}_1 = (0,1,0), \mathbf{x}_2 = (0,0,1), \mathbf{x}_3 = (1,0,1), \mathbf{x}_4 = (1,1,1)$$

has three control sequences and

$$\mathbf{x}_0 = (1,0,0), \mathbf{x}_1 = (1,1,0), \mathbf{x}_2 = (0,1,1), \mathbf{x}_3 = (1,0,1), \mathbf{x}_4 = (1,1,1)$$

has one control sequence.

**Example 3.** We consider a larger example with 15 nodes and three controls from [16] modeling *Drosophila melanogaster*. The dynamics in simplified notation are in Table 1. The initial state is $\mathbf{x}_0 = (0,1,1,0,0,1,0,0,0,0,0,0,1,0,1)$ and the desired target is $\mathbf{x}_5 = (0,0,0,1,1,0,1,0,0,0,1,0,0,0,0)$ in M=5 steps.

The polynomial method finds trajectories with controls from $\mathbf{x}_0$ to $\mathbf{x}_5$, such as:

| node | logical rule for update |
|---|---|
| SLP | SLP |
| wg | ((CIA & SLP & !CIR) \| (wg & (CIA \| SLP) & !CLR)) & U2 |
| WG | wg |
| en | !SLP |
| EN | en |
| hh | EN& !CIR & U3 |
| HH | hh |
| ptc | CIA & !EN & !CIR & U1 |
| PTC | ptc & PTC |
| PH | PTC |
| SMO | !PTC |
| ci | !EN |
| CI | ci |
| CIA | CI&SMO |
| CIR | CI &! SMO |

Table 1: Boolean model of *Drosophila melanogaster* dynamics from [16]

$\mathbf{x}_1 = (0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1)$,
$\mathbf{x}_2 = (0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0)$,
$\mathbf{x}_3 = (0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0)$,
$\mathbf{x}_4 = (0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0)$,
$\mathbf{x}_5 = (0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0)$,
with controls $\mathbf{u}_0 = \mathbf{u}_1 = \mathbf{u}_2 = \mathbf{u}_3 = (1, 1, 1)$ and $\mathbf{u}_4 = (1, 1, 0)$.

**Example 4.** Stochastic transformations are controlled in [15]. We look at their Example 2 assuming independence in the selection of $\tau = 3$ transformations (their process is Markovian). The target set $A$ is defined by `M=0`, with ideal $I_A = \langle t_1 \rangle$ and the three transformations are in Table 2.

| node name | logical rule for update |
|---|---|
| $M$ | $!G_e$ & $(L \mid L_m)$ |
| $L$ | 0 |
| $L_m$ | 0 |
| $M$ | $!G_e$ & $(L \mid L_m)$ |
| $L$ | 0 |
| $L_m$ | $M$ & $!G_e$ |
| $M$ | $!G_e$ & $(L \mid L_m)$ |
| $L$ | $M$ & $!G_e$ |
| $L_m$ | $!G_e$ |

Table 2: Three Boolean models for stochastic dynamics from [15]

One finds that $B_1$ contains all states, and the control $G_e$ (as $u_0$) with $G_e = 1$ will send $\mathbf{x}_0 = (1, 1, 1)$ to $\mathbf{x}_1 = (0, 0, 0)$ in one step, for any of the three transformations.

**Example 5.** We consider a model built with statistical learning methods on longitudinal data from in-home motion sensor recordings first given in [7]. The data was generated by the Intelligent Systems for Assessing Aging Changes (ISAAC) study [12], an aging study conducted by ORCATECH at OHSU. The target set $A$ is where `mci`=0, a condition of no cognitive impairment. Possible controls are labeled as $u_1, u_2$ in Table 3, mean walking speed and number of walks, which may be possible to adjust with a training program.

| | |
|---|---|
| numtrans | number of transitions between rooms, normalized to mean 1.0 in home |
| numfir | number of sensor firings in the home, normalized to mean 1.0 in home |
| oohhours | estimate of the number of hours the participant spent out of home |
| timeasleep | total time asleep (minutes) |
| ttib | total time in bed (minutes) |
| waso | time awake after sleep onset (minutes) |
| sleeplatency | total sleep latency (minutes) |
| sleeplivroom | time asleep in living room (minutes) |
| compuse | total computer use (minutes) |
| numwalks | number of walks (count) |
| meanws | mean walking speed (cm/s) |
| wsq3 | upper quartile of walking speed (cm/s) |
| wscv | coefficient of variation of walking speed |
| wssigma | standard deviation of walking speed (cm/s) |
| mci | mild cognitive impairment diagnosis with the Jak criteria (binary indicator) |

Table 3: ISAAC Study Variable Definitions

One finds that the controllable set for ideal $B_2$ is all configurations. An initial point with `wsq3`=0, `compuse`=0, and `waso` = 1 will move to `mci=1` in the next step, regardless of the controls or the starting value of `mci`. In two steps however we can move the initial state (1,0,0,0,0, 0,0,0,0,1, 1,0,0) (with initial conditions `mci=1`, `sleeplivroom`=1, `waso`=1 showing sleep disorder) to the target set with controls $\mathbf{u}_0 = (1,1), \mathbf{u}_1 = (1,1)$, a rigorous walking regimen. The model is built with observational data so any regimen is only a possibility.

| | |
|---|---|
| | targets, functions |
| 1 | mci, !wsq3 & !compuse & (!waso & sleeplivroom \| waso) |
| 2 | numtrans, numfir |
| 3 | numfir, !numtrans & (!sleeplivroom & !wsq3 \| sleeplivroom) \| numtrans |
| 4 | oohhours, !ttib & (!sleeplivroom & (numfir & numwalks \| !numfir) \| sleeplivroom) |
| 5 | wsq3, !meanws & !numwalks & !mci \| meanws |
| $u_1$ | meanws, wsq3 & (wscv & wssigma \| !wscv) |
| $u_2$ | numwalks, wssigma & !wscv |
| 6 | wscv, wssigma & (!meanws \| meanws & mci ) |
| 7 | ttib, timeasleep |
| 8 | wssigma, wscv & meanws |
| 9 | timeasleep, ttib |
| 10 | sleeplivroom, timeasleep & !ttib |
| 11 | waso, sleeplatency & ttib |
| 12 | compuse, !mci & wssigma |
| 13 | sleeplatency, waso & ttib |

Table 4: Life Kinetics Dynamics in Logical Notation

# 5 Discussion

The method of control with polynomial algebra presented in Sections 2 and 3 is practical on standard examples of Boolean network control. An advantage is the clarity and completeness of the algorithm. A question is whether it can handle larger networks, which challenge existing methods. In fact commutative algebra may be able to handle some large problems because it is not hindered by the $2^d$ cardinality of the state space as some methods are. For example, the polynomials needed to code the binary space $\{0,1\}^d$ number just the logarithm $d$, and relevant sets in Boolean network control study are sometimes only slightly more complicated. The hard operation in the method is variable elimination (as in $I \cap K[\mathbf{s}]$, eliminating other variables like $\mathbf{t}$ and $\mathbf{v}$), which uses Groebner basis foundations. The complexity bounds as in [10] are not promising but most problems are much easier than the worst-case bound suggests. On all the examples considered here, the calculations were completed almost instantly on an ordinary computer. Also, the polynomial methods generalize easily to more general state spaces.

**Acknowledgment.** This research was partially supported by NIH grants P30AG066518 and P30AG008017.

# References

[1] Abbott, J., and Bigatti, A. M. (2014). *CoCoALib: a C++ library for doing Computations in Commutative Algebra*. Available at http://cocoa.dima.unige.it/cocoalib.

[2] Bell, E. (1927). Arithmetic of Logic. *Transactions of the American Mathematical Society, 29*(3), 597-611.

[3] Cheng, D., Qi, H., and Li, Z. (2011). *Analysis and Control of Boolean Networks, A Semi-Tensor Product Approach.* London, Springer-Verlag.

[4] Cox, D., Little, J., and O'Shea, D. (1997). *Ideals, Varieties, and Algorithms 2nd Edition.* New York, Springer-Verlag.

[5] Decker, W., Greuel, G.-M., Pfister, G., and Schönemann, H. (2020). SINGULAR 4-2-0 — A computer algebra system for polynomial computations. Available at http://www.singular.uni-kl.de.

[6] Dinwoodie, I. H. (2014). Conditional tests on basins of attraction with finite fields. *Methodology and Computing in Applied Probability, 16*(1), 161-168.

[7] Dinwoodie, I. H. (2016). Computational methods for asynchronous basins. *Discrete and Continuous Dynamical Systems B, 21*(10), 3391-3405.

[8] Grayson, D. R., and Stillman, M. E. (2014) *Macaulay2, a software system for research in algebraic geometry.* Available at http://www.math.uiuc.edu/Macaulay2/.

[9] Gross, E., Harrington, H., Meshkat, N., and Shiu, A. (2020). Joining and decomposing reaction networks. *Journal of Mathematical Biology 80*(1), 1683-1731.

[10] Hashemi, A., and Lazard, D. (2006). Complexity of 0-dimensional Gröbner Bases. RR-5660, INRIA.

[11] Jacob, F., and Monod, J. (1961). Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology 3*(3), 318–356.

[12] Kaye, J. A., Maxwell, S. A, Mattek, N., Hayes, T. L., Dodge, H., Pavel, M., Jimison, H. B., Wild, K., Boise, L., and Zitzelberger, T. A. (2011). Intelligent systems for assessing aging changes: home-based, unobtrusive, and continuous assessment of ageing. *The Journals of Gerontology, Series B: Psychological Sciences and Social Sciences, 66B*(S1) i180-i190.

[13] Klarner, H., Heinitz, F., Nee, S. and Siebert, H. (2020). Basins of Attraction, Commitment Sets, and Phenotypes of Boolean Networks. *IEEE/ACM Transactions On Computational Biology And Bioinformatics, 17*(4), 1115-1124.

[14] Kreuzer, M., and Robbiano, L. (2000). *Computational Commutative Algebra I.* New York, Springer-Verlag.

[15] Meng, M. , Xiao, G., Zhai, C., and Li, G. (2019). Controllability of Markovian jump Boolean control networks. *Automatica, 106*, 70-76.

[16] Moradi, M., Goliaei, S., and Foroughmand-Araabi, M.-H. (2019). A Boolean network control algorithm guided by forward dynamic programming. *PLoS ONE, 14* e0215449. doi:10.1371/journal.pone.0215449.

[17] Thomas, R. (1973). Boolean formalization of genetic control circuits. *Journal of Theoretical Biology, 42*(3), 563-585.

[18] Zhao, Y., Ghosh, B., and Cheng, D. (2015). Control of Large-Scale Boolean Networks via Network Aggregation. *IEEE transactions on neural networks and learning systems, 27*(7), 1527-1536.

[19] Zhegalkin, I. I. (1927) On the technique of calculating propositions in symbolic logic. *Matematicheskii Sbornik, 34*(1), 9-28.