Tutorial: Pneumatic Sequencing (II)

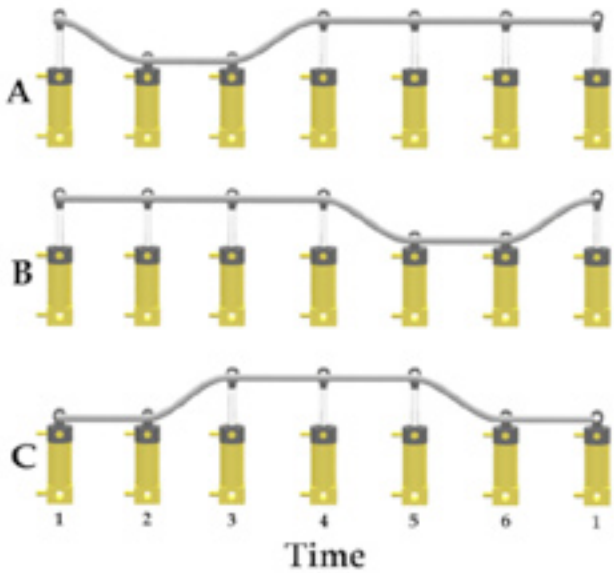
In this second part of the tutorial you will see a practical application of e pneumatic sequence and how to solve problems in a circuit with the help of diagrams..

By Kevin Clague
Edited by Jetro de Château

Three pistons Five switches

I wanted to build a four legged pneumatic walker (Quad242) to show off a creation of mine, a pneumatic polarity reverser. I thought the reverser was the fancy part, but in fact making the walker walk the way I wanted was the hard part. I learned much of what I know about pneumatic sequencers making Quad 242 walk.

Quad242 has four feet split up into two synchronized pairs. The front left and back right feet do the same thing, and the front right and back left feet do the same thing. Quad242 walks by driving its feet back, lifting them off the ground, bringing them forward and dropping them on the ground, over and over. I wanted to make Quad242 put the up feet down, before it moved the down feet up, so it would have two feet down, then four feet down, and then two feet down (thus the name Quad242). To make this happen I needed a pneumatic circuit that created this waveform.



Piston A represent the front left and back right vertical leg movement (expanded pistons are on the ground). Piston B represents the front right and back left vertical leg movement. Notice that in the first step both pistons A and B are expanded. Piston A then contracts and expands leaving both piston A and B expanded. Piston B then contracts and expands leaving both piston A and B expanded and the cycle repeats. Piston C is used to control both piston A and piston B.

Circuit Analysis and Derivation

First pass analysis of the waveform indicates that at time 1, piston B has just expanded, which makes piston A contract. This gives us the formula:

Ac = B

At time 2, piston A just contracted which makes piston C expand. This gives us:

Cx = ~A

At time 3, piston C completes expanding which makes piston A expand, so:

Ax = C

At time 4, piston A completes expanding, which makes piston B contract:

Bc = A

At time 5, piston B completes contracting, which makes piston C contracts so:

Cc = ~B

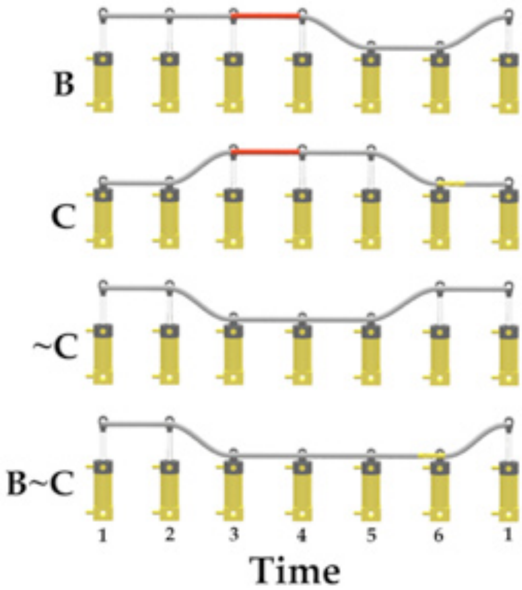
At time 6, piston C completes contracting, which makes piston B expands, so:

Bx = ~C

Now we need to examine piston A's formulas for pressurization.

Ax = C
Ac = B

If you pressurize both ports of a pneumatic piston at the same time, the piston's behavior is unpredictable. Your pneumatic sequencer will probably lock up and stop sequencing. To make sure that this won't happen in our circuit, we need to examine the inputs to each piston over time. This timing diagram represents the pressures going into piston A from switch ports B and C over time.



Notice that B and C are both providing pressure to piston A in time 3 (indicated by red hose). To avoid this we need to modify one of the two formulas. C starts providing pressure into Ax at the right time, and B starts providing pressure into Ac at the right time, but the pressure from B stays on and overlaps the pressure from C. We can modify the pressure into Ac using an and gate by adding another switch to piston B. Since Ax's pressure is behaving correctly and Ac's pressure is behaving incorrectly, we need to modify Ac's formula. If we and ~C (because Ax = C) with B, by running B through the new switch on piston C, we get a new Ac that does not overlap with Ax and still contracts piston A at the right time.

This analysis leads to these formulas for piston A:

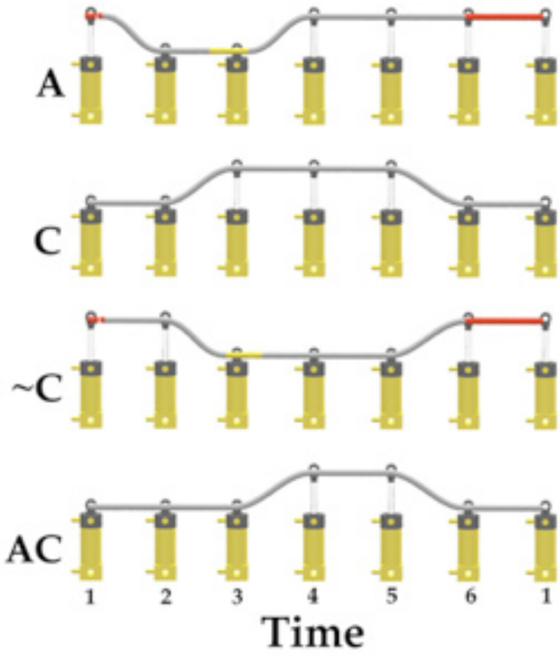
Ax = C
Ac = B~C

Note that at time 6, piston A is depressurized indicated by the yellow hose.

Piston B's initial expansion and contraction formulas are:

Bx = ~C
Bc = A

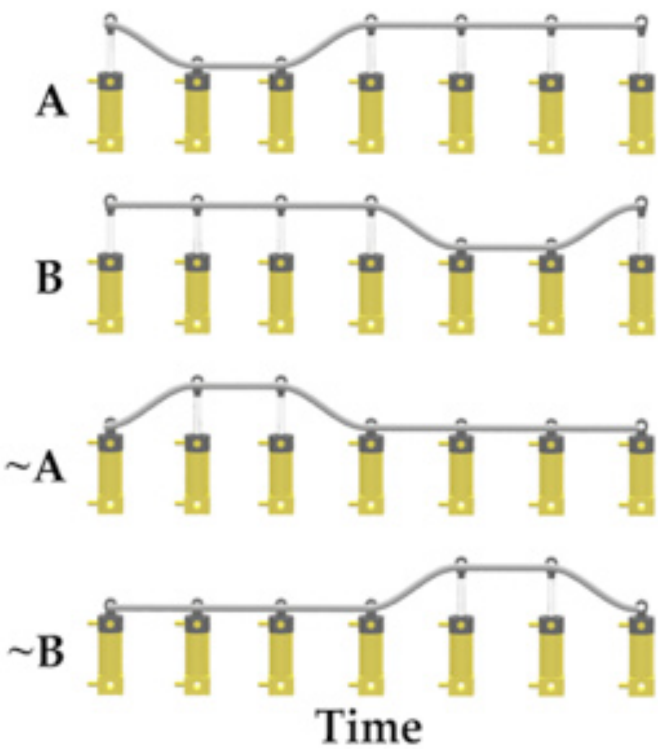
Examining Piston B's pressure ports graphically reveals another double pressure point in the cycle in time 6, and time 1. Bc's pressure stays on too long, so we need to modify Bc's formula. We can resolve piston B's double pressure problem by adding another switch to piston C, and run A through it creating the formula Bc = AC. AC pressurizes piston B at the right time, but does not overlap with ~C. Notice that piston B is not pressurized on either port during time 3.



Piston B's final formulas are:

Bx = ~C
Bc = AC

Piston C is expanded by ~A and contracted by ~B. Studying these waveforms show that piston C has no pressure conflicts, so the initial formulas remain unchanged. Note that piston C is depressurized at two points in the cycle.



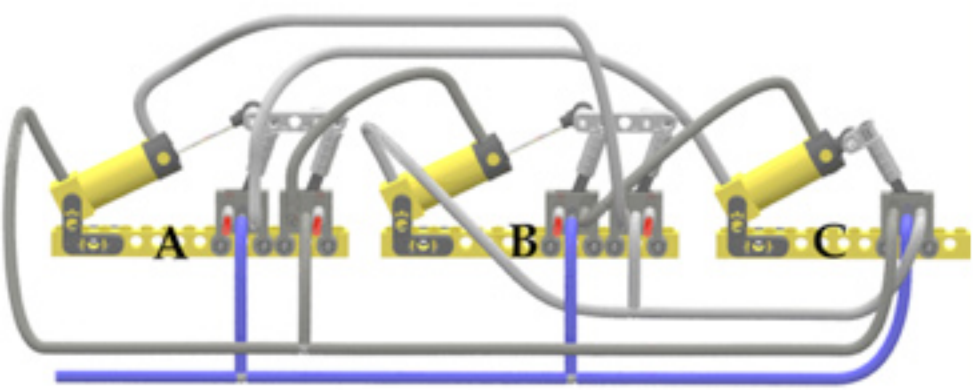
After simultaneous pressure analysis and avoidance, we end up with these equations describing out circuit:

Ax = C
Ac = B~C

Bx = ~C
Bc = AC

Cx = ~A
Cc = ~B

Here is a diagram of circuit 7:



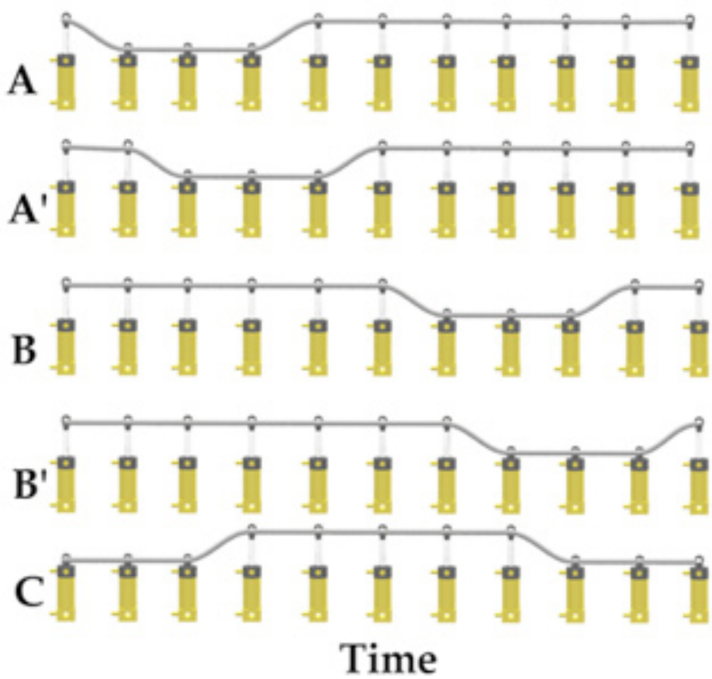
Outside Forces

Circuit 7 is the base circuit used in my four legged walker named Quad 242. One pair of feet is controlled by piston A, and the other by piston B. Piston C controls when the legs go forward and backward.

When I first designed Quad242, I used circuit 7 to control it. I added synchronized twin pistons for piston A and piston B, so that there were four vertical pistons, one for each leg. I also added three more copies of piston C (one for each leg), that were synchronized with piston C. The circuit did not function as desired, because pistons A and pistons B are depressurized during the six step sequence. This made Quad242's body drop to the floor twice per cycle due to gravity pulling the body downward. I tried many ways to make piston A be pressurized at every point in the sequence, with each attempt ending in failure.

Finally I decided that piston A could not be weight bearing. I realized that I could create an always pressurized copy of A and an always pressurized copy of B by making these simple additions to the circuit.

The wave form for the new circuit looks like this:



Piston A’ (pronounced A prime) is driven by piston A’s switch, so A’ does what A does, except delayed in time. Piston B’ is driven by piston B’s switch, so B’ does what B does, except delayed in time. The output from piston A’ is used to control the other pistons, instead of A, and likewise for B’ and B. This means that A’ and B’ are synchronized with the circuit, but are always pressurized (which means they can be weight bearing.)

The formulas that describe circuit 8 are:

Ax = C
Ac = B’~C

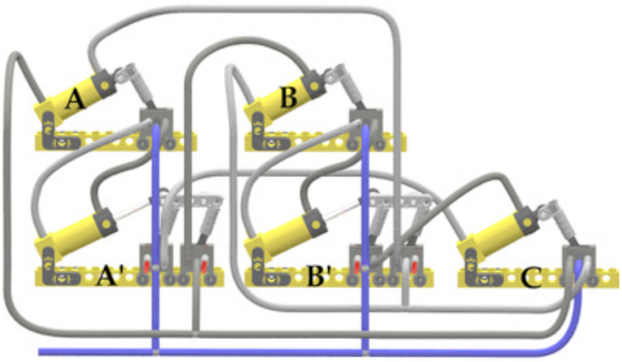
A’x = A
A’c = ~A

Bx = ~C
Bc = A’C

B’x = B
B’c = ~B

Cx = ~A’
Cc = ~B’

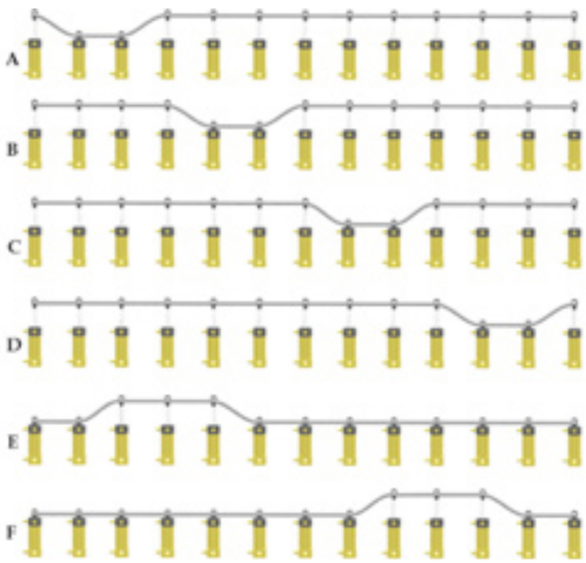
This gives us this circuit diagram:



Piston C is not a weight bearing piston, so we do not need to add an extra always pressurized version of piston C.

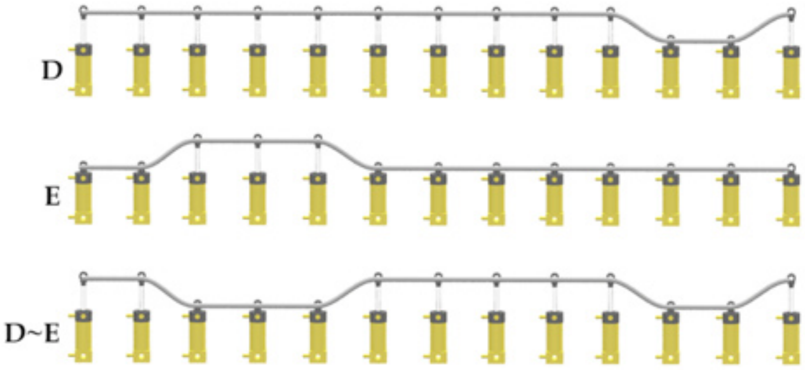
Unachievable sequences

One of my next projects is a four legged walker that always has at least three feet down. When transferring weight it would have all four feet down. It will probably be named Quad343. I tried to create this sequence for Quad343.



That's a lot of sequence!

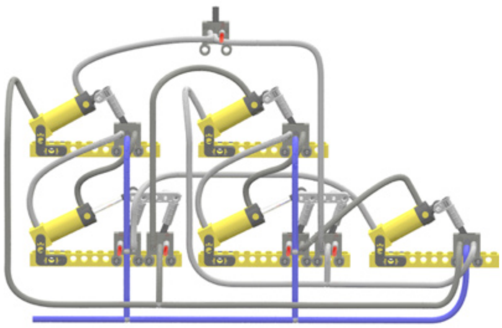
I started to do analysis for piston A which contracts when D expands, and expands when E expands, giving this pressure diagram:



The first problem is that D and E are both pressurizing A at the same time. When trying the standard trick of and-ing the one that is on too long with the not of the one that is being trashed, it doesn't work. The last row of the diagram shows the result. The problem is that D~E pressurizes A at two different points in the cycle. Not good because piston A will expand/contract twice in the same cycle violating the original goal for A. The result is that we cannot create a circuit to generate the wave form as proposed. I'll have to come up with a different sequence for Quad343 (I've already got that one figured out :^)

Stopping and Starting

It is possible to make your sequencer stop and start under external control. If you disconnect a pressure port to a piston, run the pressure through a stand alone switch, and hook the output of the switch back to the piston, you can make it start and stop. By preventing the pressure from hitting the piston, you prevent it from expanding or contracting (depending on the port you modify). Preventing the piston from changing prevents the circuit from making forward process. This is circuit 8 with a switch that can freeze the sequencer at the beginning of the sequence.



If your sequencer is part of a carnival ride that changes shape over time, you might want to stop and start the ride so your mini-figs can get on and off. Stopping the sequencer at the end of the ride could be controlled by an RCX that controls a motorized pneumatic switch.

Your complete model could very well contain two sequencers that interact and stay coordinated by controlling each other's forward progress.

You can put more than one stopping point switch into your sequencer so you can stop at multiple points in the cycle.

Each of the stopping point switches acts as an and gate that controls forward progress.

Where to go from here

There are any number of possible places to use complex pneumatic sequences:

- Animated creatures walking, or moving appendages
- Carnival rides that change shape over time
- Simulation of manufacturing processes (a common use of pneumatics in real life)
- Pneumatic computing devices

Summary

This tutorial explained how to create pneumatic circuits that go through repeating sequences of pneumatic piston expansion and contraction. It explained how to make multiple pistons be coordinated (synchronized) even though the pistons themselves expand or contract at different rates. It explained how to start from a description of the desired sequence and see if a circuit can be created for it. This chapter introduced the concepts of boolean logic, how it is implemented in LEGO® pneumatics, and how it can be used to create pneumatic sequencers.

Pneumatic sequencers can be used to create complicated and advanced self actuated LEGO models that require no motors or RCX.
#