In the Spotlight: Verifying Automated Theatrical Follow-Spots

Becky Jesurum

Abstract

Automated spotlights are one of the next steps in the automation of theatrical lighting. I have developed a hybrid system to model the control and motion of such a light following a single actor, and endeavored to ensure that the light has the capability to both move smoothly and stay on the actor at all times. My findings are presented in this paper.

1 Introduction

In theatrical and other staged performances, most lighting is automated. Predetermined cues are programmed into a board, and during the show an operator will activate each cue and the lights will automatically change. Automated lighting is practically universal, because it is extremely reliable and allows for much more complicated lighting plots and transitions than a human alone could execute. Automated lights consist of both stationary fixtures, which give control over light intensity (and sometimes color), and moving "intelligent" fixtures, which can have complex movements programmed, as well as changes in beam width and shape. One prominent lighting type that is not automated is the follow-spot light.

The purpose of a follow-spot is to specifically track one actor on the stage. They are widely used during concerts and musicals to draw focus to the actor singing, illuminating them against the background. Unlike conventional lighting fixtures, spotlights need to have continuous information of an actors' location. Historically, this meant that human spot-operators would control the spotlight's motion. This is ideal, as not only can the operators follow the actors as they move, but they can anticipate where the actor will next go based on prior experience. Automated spotlights have started to find their way into industry, though. The two primary systems right now are Wybron's AutoPilot II [2] and BlackTrax [1] (which is used by Cirque de Soleil). These systems are not very popular because as of yet, a human will still typically perform the job better. However, as technology continues to advance, these products will improve and become used more frequently. Proving the safety of automated follow-spot lights could be a positive step in their widespread acceptance.

In this paper, I outline and prove the safety of a model of an automated spotlight. I model a simple light following a single actor on an infinite stage. I do not have access to the actual control decisions that the existing systems make, but the proof of the simplified model indicates that there are valid control decisions for an automated follow-spot system.

Figure 1: A regular follow-spot light

Figure 2: A spot-operator

2 Related Work

As noted in the introduction, automated spotlight technology is currently in development by a few companies. BlackTrax and AutoPilot II are both described at high-level in their advertisements, but the technical details are left out, as they are trying to sell their unique software and hardware, not publish it.

The problem of designing a model is similar to that of determining adaptive cruise control [3] in the presence of other cars. In both cases, the controlled entity moves autonomously relative to another entity, trying to stay within (or without) some range of distance.

3 Model Description and Assumptions

The goal of the light is to stay focused on a specific actor, so the model only needs to consider the light and the actor. Furthermore, the actor generally has a transmitter and the light a receiver, so it realistic to assume the light knows the actor's position The model will start in one dimension, and then increase to two, and possibly three dimensions.

3.1 Desired Properties

At a minimum, the light should always completely encompass the actor. It should always move smoothly, as a jerky light would detract from the actor. The light should also be as centered on the actor as possible, but this is not critical. In other words, the center of the light should be as close to the center of the actor as possible.

3.2 Actor

3.2.1 Representation

The actor will be represented as a ball of some fixed radius. An actual actor would be more oblong, but approximating him as a ball (with width equal to his height) simplifies computation. This is justified because if the light covers the entire ball, then it will certainly cover the strictly smaller actor.

3.2.2 Motion

As actors move at walking speeds, they typically have the ability to change their velocity at will. For the sake of the model, the actor will have discrete control over his velocity and during each time step will not accelerate. This assumption is made because the time step such a light can achieve is under half a second, which is typically less time than a person will use between subsequent changes in velocity.

3.3 Light

3.3.1 Representation

Spotlights typically form a circle around their target, so the illuminated region of the light (henceforth just referred to as light) will also be modeled as a ball with a radius strictly greater than that of the actor (the area of a spotlight is usually far larger than that of the actor it is trained on).

3.3.2 Motion

In a realistic light fixture, the beam of light moves via angular rotation of the chassis. Mathematically modelling angular motion is complicated, so to simplify the model, we assume that the light beam moves with Cartesian motion. Since at large distances, tangential velocity and acceleration are far larger than their angular counterparts (small change in angle is a large change in linear distance), we will also assume that the light can move and accelerate with far greater magnitudes than the actor.

The light will have access to the actors' position at each time step. Unlike the actor, the light will not have discrete control over its velocity (instead, its velocity will only be effected as a result of the light's constant acceleration). This is to better ensure that it moves smoothly.

3.4 Environment

The model will exist in an infinite space, ignoring the existence of a stage. If the light can follow the actor anywhere, then it can certainly do so within the confines of a stage. Conceptually, acknowledging the existence of a stage better justifies the use of Cartesian movement for the light: the further out a light rotates, the more distorted and weaker its beam becomes. Furthermore, the signal from the actor to the light weakens as distance increases. Within a fixed bound, the distortion will be minimal, allowing the Cartesian assumption. Similarly, within the confines of a stage, the signal from the actor should be relatively uniform.

The model will be time-triggered, as the actual spotlight periodically receives transmissions from the actor. The alternative would be an event-triggered model, in which whenever the actor changes his velocity, the light is alerted to also change its motion. The time-triggered model better reflects reality in this case.

4 Formal Model

4.1 Outline

A program can be modeled like so: $Pre \rightarrow [(Control; Physics)^*@(invariant)]Post$. This means that if *Pre* holds, then after running *Control* and *Physics* sequentially for a non-deterministic number of repetitions, *Post* will always be true. *invariant* must be true before the loop executes, and after each run of the loop.

4.2 Program Constants

We represent the maximum time between each transmission with `T`. The maximum velocity allowed to the actor is `V`. The radii of the light and actor are `r` and $\mathtt{r}_a$, respectively. All of these constants must be positive.

Furthermore, we will require that `V*T` $<$ `r`. This is a limitation on how far the actor can physically move in a time step. Given actual stage conditions, this is a reasonable bound. The average actor will not exceed speeds of a quick jog, approximately $3m/s$, and an average sensor might take up to $.5s$ to receive and transmit its data. $V * T = 3m/s * .5s = 1.5m$, which is less than the average height of an actor, and therefore less than the radius of the beam. This property may not be explicitly used in the current proof, but is a good bound on the actor's velocity. The velocity of the light should be bounded, too, but this model does not do that. In this case, this restriction would be very important.

4.3 One Dimension

In one dimension (we will imagine the x-axis), the actor and light each have a single position coordinate (x), a velocity (v), and an acceleration (a). To keep consistent with the naming convention used for the radii, the light will have position, velocity, and acceleration `x`, `v`, `a`, and the actor $\mathtt{x}_a$, $\mathtt{v}_a$. To track the current time in a time cycle, we also include the variable `t`.

Figure 3: The one-dimensional model. Red represents the light and blue the actor. `x` and $\mathtt{x}_a$ are the centers and `r` and $\mathtt{r}_a$ the radii of the light and actor respectively.

4.3.1 Preconditions

$Pre \equiv$ `T > 0 & V > 0 &` $\mathtt{r}_a$ `> 0 & r >` $\mathtt{r}_a$ `& V*T > r &` *light_on_actor*

As discussed above, all constants should be positive, and the radius of the light should be strictly greater than that of the actor. Furthermore, the light should be on the actor at the beginning (ideally when the spotlight turns on, it is on target already). *light_on_actor* is an alias for a block of code. To be true, the light must overlap the entire actor. Referencing Figure 3, it is clear that this property is true if the distance between the centers plus the radius of the actor is no greater than the radius of the light.

light_on_actor $\equiv \mathtt{r}_a + \mathtt{abs(x - x}_a\mathtt{)} \leq \mathtt{r}$

4.3.2 Control

$Control \equiv Environment; Actor; Light$

As was hinted at above, a semi-colon indicates sequential composition of different programs. There are three elements of control in this system, which will be further detailed below.

Environment
The only environment control is time, and the time variable must be reset to 0 for each time-step. This is because Physics runs for `T` time by setting the time derivative of `t` to `1`.
Environment $\equiv$ `t:=0`

Actor
The actor gets discrete control over velocity, with an upper bound. This is because at any point in time, the actor is able to choose his direction and speed (within the bounds of what is realistic — he cannot physically move at 60mph for instance). An asterisk (`*`) non-deterministically picks a value. The test operator `?P` checks that `P` is true, and aborts otherwise. In this case, this has the effect of only allowing values within the boundary range for velocity.
Actor $\equiv \mathtt{v}_a := *;\ ?(\mathtt{v}_a \leq \mathtt{V}\ \&\ \mathtt{v}_a \geq \mathtt{-V});$

Light
The light has discrete control over its acceleration, and can only choose accelerations that

will keep it on the actor. One option here is to choose a specific acceleration that we know will work — for instance one which will the light's location after a time step be the actors' current location. However, this severely limits the model. Instead, we should allow for many different accelerations and support that the light will never leave the actor. This model is more general and therefore more useful, as it justifies the safety of numerous different control choices, instead of just one.
$\mathit{Light} \equiv \texttt{a:=*; ?}(\mathit{safe\_acc})\texttt{;}$

Like *light_on_actor*, *safe_acc* is an alias for more code. To write *safe_acc*, we need to augment our definition of *light_on_actor* to allow for input values, rather than using the current values for $\texttt{x}$ and $\texttt{x}_a$.
$\mathit{light\_on\_actor}(x, x_a) \equiv \texttt{r}_a + \texttt{abs}(x - x_a) \leq \texttt{r}.$

To know that our acceleration is okay, we need to know that at the maximum distance between the light and the actor within a time cycle $\texttt{T}$, the light will remain on the actor. The extreme values to be checked are at the beginning and end of the time cycle, and any local maxima and minima (Figure 4).

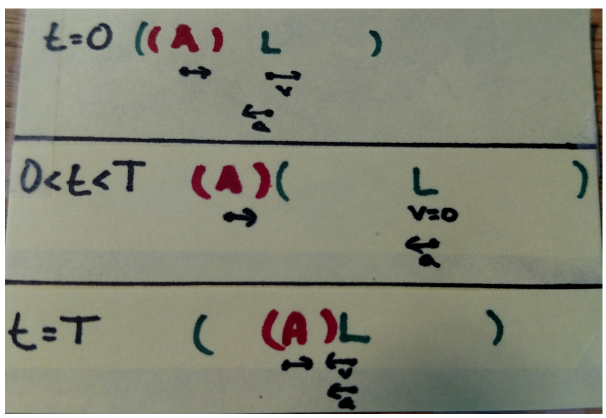

Figure 4: A sample scenario illustrating the need to check the local max/min point. At the start, the actor has chosen his new rightward velocity, the light is still moving right, and the light has chosen a (unsafe) leftward acceleration. When the light's velocity switches signs, it is no longer covering the actor! However, then once the time cycle is complete, the light is on the actor again. Only checking the start and the end conditions would in this case allow for an unsafe spotlight.

$\mathit{safe\_acc} \equiv \mathit{safe\_at\_end}$ & $\mathit{safe\_max\_dist}$

Since the light does not know the actors' velocity, it examines the edge cases. Whether the actor is moving as positively or as negatively as possible, the light must remain on him at the end of the time cycle.

$$
\begin{aligned}
safe\_at\_end \equiv\ & light\_on\_actor(\mathtt{x + vT + }\tfrac{1}{2}\mathtt{aT}^2\mathtt{,\ x}_a\mathtt{\ + VT)} \\
\&\ & light\_on\_actor(\mathtt{x + vT + }\tfrac{1}{2}\mathtt{aT}^2\mathtt{,\ x}_a\mathtt{\ - VT)}
\end{aligned}
$$

The distance between the actor and light at a given time t is $\mathtt{x + v*t + }\frac{1}{2}\mathtt{a*t}^2\mathtt{ - (x}_a\mathtt{ + v}_a\mathtt{*t)}$. We take the first derivative and set it to zero to find the local extrema. $\mathtt{v + at - v}_a\mathtt{ = 0}$, giving that $\mathtt{t} = \frac{\mathtt{v}_a - \mathtt{v}}{\mathtt{a}}$. So, again we look at the edge cases for the actors velocity, but instead of time T, at the newly derived time.

$$
\begin{aligned}
safe\_max\_dist \equiv\ & light\_on\_actor(\mathtt{x + v}\tfrac{\mathtt{V - v}}{\mathtt{a}}\mathtt{ + }\tfrac{1}{2}\tfrac{\mathtt{V - v}^2}{\mathtt{a}}\mathtt{,\ x}_a\mathtt{\ + V}\tfrac{\mathtt{V - v}}{\mathtt{a}}\mathtt{)} \\
\&\ & light\_on\_actor(\mathtt{x + v}\tfrac{\mathtt{-V - v}}{\mathtt{a}}\mathtt{ + }\tfrac{1}{2}\tfrac{\mathtt{-V - v}^2}{\mathtt{a}}\mathtt{,\ x}_a\mathtt{\ - V}\tfrac{\mathtt{-V - v}}{\mathtt{a}}\mathtt{)}
\end{aligned}
$$

4.3.3 Physics

We assume that both the actor and light move according to Newtonian dynamics. Thus, position varies with velocity, and (if applicable) velocity with acceleration.
$Physics \equiv \{\mathtt{x}_a\mathtt{' = x}_v\mathtt{,\ x' = v,\ v' = a,\ t' = 1\ \&\ t} \leq \mathtt{T}\}$

This ODE will run for no more than T time, emulating the physical restraint on how often the light can make a control decision. Sadly, it limits the actors' control decision to this time constraint also.

4.3.4 Invariant and Postcondition

The postcondition for the problem is just that the light is on the actor.
$Post \equiv light\_on\_actor$

The invariant must follow from the precondition, imply the postcondition, and be true after each run of the loop containing the controls and dynamics. Since the *light_on_actor* is inductive, it is sufficient as an invariant.
$invariant \equiv light\_on\_actor$

4.4 Two Dimensions

For two dimensions, we must augment the model. Now, the position and motion of the light and actor have y components as well as x components. Now, *light_on_actor* can be redefined based on the distance between the centers of the actor and light.
$light\_on\_actor \equiv \mathtt{(x - xa)}^2\mathtt{ + (y - ya)}^2 \leq \mathtt{(r - ra)}^2$.

As in one dimension, the challenge of this model is ensuring that the light chooses a safe acceleration. Here we cannot just plug in the maximum velocity of the actor in each direction, as there are an infinite number of directions to choose from. I have yet to determine an efficient method by which to ensure the safety of acceleration for this model.

5 Proof Techniques

5.1 One Dimension

In actually writing the model, every instance of `abs(x)` $\leq$ `b` is replaced with `x` $\leq$ `b | x` $\geq$ `-b`. Alternatively, this could have been solved by squaring both sides of the inequality ($\mathtt{x}^2 \leq \mathtt{b}^2$). This proof depends greatly on the loop invariant and $safe\_acc$ defined above. In applying the loop invariant rule (Figure 5), the problem simplifies into showing that the loop invariant holds after an iteration of the program. The justification for this comes largely from the safe acceleration condition.

$$\text{loop}\ \frac{\Gamma \vdash J, \Delta \quad J \vdash [a]J \quad J \vdash P}{\Gamma \vdash [a^*]P, \Delta}$$

Figure 5: The loop invariant rule in KeYmaera X. If the preconditions imply the loop invariant, the loop invariant is true after every run of the loop, and the loop invariant implies the postcondition, then the postcondition holds after running the initial program. This is a very powerful rule, as it eliminates nondeterministic repetition.

$safe\_acc$ along with our preconditions provides support that all extreme values of the distance between the actor and light are safe (extrema being the endpoints and any local extrema). It then logically follows that any intermediate values will also be safe, satisfying the proof.

6 Conclusion

The results indicate that at least in one dimension, the safety of an automated spotlight can be proved. The goal was to demonstrate a safety proof (that is, that the light will always remain on the actor) for both one and two dimensions, but technical difficulties with the absolute value function in KeYmaera X caused a significant delay in progress toward that end.

For future work, much can be done. The proof of a two dimensional model is the first step. Another step that could be taken are to weaken or remove some assumptions that were made. This could include allowing the actor to accelerate, not assuming that the light moves via linear kinematics (accounting for how rotation affects linear speed and acceleration, as well as beam shape and size), and not assuming perfect communication between the light and the actor (real transmitters sometimes communicate erroneous or otherwise inaccurate values). As is, this model is just complex enough to be useful, but more robust models could provide significantly more confidence in the effectiveness of these lights.

7 Model and Proof Tactic: One dimension

```
Functions.
  R T().  /* global time-step */
  R V().  /* maximum actor velocity (magnitude) */
  R ra(). /* actor radius */
  R r().  /* light beam radius */
End.

ProgramVariables.
  R x.    /* light center */
  R v.    /* light velocity */
  R a.    /* light acceleration */
  R xa.   /* actor center */
  R va.   /* actor velocity */
  R t.    /* time */
End.

Problem.
/* Preconditions: ensure positive constants and light_on_actor */
(T > 0 & V > 0 & ra > 0 & r > ra & V*T < r
  & x - xa <= r - ra & x - xa >= ra - r)
->
[{
  /* Discrete dynamics */
  t := 0; va := *; ?(va <= V & va >= -V);   /* reset time, set actor speed */
  a := *; ?(                                /* set light acceleration */
  /* ensure light_on_actor after T time */
    (x + v*T + a*T^2/2) - (xa + V*T) <= r - ra
   &(x + v*T + a*T^2/2) - (xa + V*T) >= ra - r
   &(x + v*T + a*T^2/2) - (xa - V*T) <= r - ra
   &(x + v*T + a*T^2/2) - (xa - V*T) >= ra - r
   /* ensure light_on_actor at maximum separation */
   &(x + v/a*(V-v) + (V-v)^2/(2*a)) - (xa + V/a*(V-v)) <= r - ra
   &(x + v/a*(V-v) + (V-v)^2/(2*a)) - (xa + V/a*(V-v)) >= ra - r
   &(x + v/a*(-V-v) + (-V-v)^2/(2*a)) - (xa + -V/a*(-V-v)) <= r - ra
   &(x + v/a*(-V-v) + (-V-v)^2/(2*a)) - (xa + -V/a*(-V-v)) >= ra - r
  );
  /* Continuous Dynamics */
  {xa' = va, x' = v, v' = a, t' = 1 & t <= T}
}*@invariant(x - xa <= r - ra & x - xa >= ra - r)
]
```

```
(x - xa <= r - ra & x - xa >= ra - r) /* post-condition */
End.


Tactic "1dimension: Proof 1".
  implyR(1) ; loop({`x-xa<=r()-ra()&x-xa>=ra()-r()`}, 1) ; <(
  master,
  master,
  composeb(1) ; assignb(1) ; composeb(1) ; randomb(1) ; allR(1) ; composeb(1) ;
  testb(1) ; implyR(1) ; composeb(1) ; randomb(1) ; allR(1) ; composeb(1) ; testb(1) ;
  implyR(1) ; solve(1) ; allR(1) ; implyR(1) ; implyR(1) ; allL({`t_`}, -11) ;
  implyL(-11) ; <(
    hideR(1) ; hideL(-8) ; hideL(-8) ; master,
    andL(-1) ; andL(-8) ; andL(-13) ; andL(-14) ; andL(-15) ; andL(-16) ; andL(-17) ;
    andL(-18) ; andR(1) ; <(
      hideL(-13) ; hideL(-14) ; hideL(-15) ; hideL(-16) ; QE,
      hideL(-12) ; hideL(-13) ; hideL(-14) ; hideL(-15) ; QE
      )
    )
  )
End.
End.
```

References

[1] BlackTrax. Tracking for lighting. http://blacktrax.cast-soft.com/tracking_lighting/.

[2] Premier Lighting. Wybron autopilot 2. http://www.premier-lighting.com/sales/autopilot.html.

[3] Sarah M. Loos, David Witmer, Peter Steenkiste, and André Platzer. Efficiency analysis of formally verified adaptive cruise controllers. *16th International IEEE Conference on Intelligent Transportation Systems*, 2013.