Project Details

Name: Speedway RFID Swimming Prototype
Researcher: Richard McCarthy
Date: 24th April 2020

Work Carried Out

Changed the planned course of work this week to address the specific MVP details of how to track the swimmer in the pool using the tag to acquire both a lap count and an approximate measurement for a lap times, relating to 50m (or 2 laps of a 25m) pool.

This solution chosen for this can have a great impact on the design of the software as it will decide the algorithm used to process the data in the standalone application and also decide the software architecture set up used in the cloud to filter the data and apply any domain specific logic to it.

RSSI

Initially a view of using the *RSSI* value (signal strength) was proposed as a way to approximate the swimmer's location when within read range. Unfortunately *RSSI* alone does not provide an accurate or more importantly reliable way to estimate distance, especially in a water environment with signal reflections off the water and continuously changing tag angles on the swimmer's head.

Extending beyond the *RSSI* measurement alone and combining it with the *Read Rate* and *Response Time* were looked at. While there may be some way of grouping data together especially with the *RSSI* and *Read Rate* to gain an insight into a pattern that could indicate an approximate position, this still suffers from the same issue of a continuously changing tag angle on the swimmer's head which will distort the values and could mislead an algorithm into thinking it was 2 meters further away than it actually is.

Changing from the front crawl stroke into either breast-stroke or buffer-fly and the situation takes a worse turn as the swimmer can be underwater for sustained periods of time.

Throughout either of these scenarios a single tag could be read (in-between signal disruptions) several hundred times while in read range facing in either direction (going up or down the pool).
Taking a very low estimate of 100 reads in the read range going both directions for 1 swimmer during a 2500m swim session.
That is 100 lengths in a 25m pool with 100 reads for each 2 lengths (both directions) which is 100 * 50 = 5000 reads. With 10 swimmers (a squad session will usually be more but keeping numbers very low) that will be 50,000 reads, at a very low end estimate.

All this data will then need to be sent to the cloud and filtered running an algorithm to find pattern groupings of data and making best guesses as to the swimmers location, which could lead to large time variances. A lap count could be deduced from this data based on a chosen read value not being within a certain assumed time to as to filter out other read values within the same lap.

Further to this as the tag is being picked up in the pool when the swimmer comes into read range and as mentioned the combination of *RSSI/Read Rates* depending on the tag angle and effects of water splashing on the signal will not provide an accurate position and not easily provide a best guess, the initial start of the first length by the swimmer may also prove problematic. The reason for this is that while a read may be picked up when the swimmer pushes off before the head goes under the water, the signal values at that point may not be a reliable reference point to use for each lap based on previously mentioned issues with tag angles and water reflection of signals.

Doppler Frequency Shift

"*The Doppler Frequency Shift is the change in frequency of a wave in relation to an observer who is moving relative to the wave source.*"

Some time was spent going through sample code with the Octane SDK. So far not found an actual programmers guide to this SDK. I have a programmers guide to the Octane LTK (low level) library so have had to gain a feel for what is really available at a more detailed level using an Integrated Development Environment (IDE) to allow quick access to all available methods in each class of code.

When running some software samples using the Octane SDK I discovered the Impinj r420 Reader is providing the Doppler Frequency Shift values along with Phase Angles.
I adjusted the sample code to output these values and ran multiple tests with both a stationary tag and a moving tag to see if these could prove useful.

The idea was that if the Doppler number was reliable that this might provide a direction of movement measurement and would also allow for the calculation of velocity of the moving tag. Given the time range from reads would be present the idea was this could provide a distance travelled from the first picked up read when the swimmer comes into range and possibly narrow down the best guess of the swimmers location.

Unfortunately this idea has been shelved on two accounts.

1 - A research paper titled "*Online People Tracking and Identification with RFID and Kinect*" from 2017 which detailed "*We observed that the Impinj R420 reader measures the Doppler shift with a standard deviation of 2.68 rad/s even when the tag is stationary. This large deviation leads to 44cm/s to 49cm/s deviation in tag velocity measurement, which is quite imprecise* "

2 - I consulted with an RFID antenna specialist via IMaR on this issue and he verified that due to the high frequencies involved, the relatively slow motion of the tag (swimmer) and all the previously mentioned issues with changing tag angles in water that the Doppler Frequency Shift readings would prove negligible

EPC UHF Gen2 Air Interface Protocol

"*EPC Gen2 Air Interface Protocol defines the physical and logical requirements for an RFID system of interrogators and passive tags, operating in the 860 MHz - 960 MHz UHF range.*"

This allows for inventory reading of tags in various "Sessions". Each EPC Gen 2 compliant tag has 2 states (A and B) for each session.

Testing of the manipulation of these sessions along with various search modes provided by the Impinj Reader was started during the initial findings phase. The location in the SDK of where to manually set this in code was found and verified with further tests this week.

The idea with this was even with the issues relating to tag angles in the water and RSSI values for approximating the swimmers location, that manipulating the tags "state" when it entered the read range might provide a marker point as such.

One of the session settings called "*Single Target with Suppression*" allows for the changing of the tag's state when it entered the read range field and keeping it surpassed (so hidden) the entire time it is within the read range field before converting the tag back to its original state (going from B to A)

This eludes to a neat way to gain an approximate marker point of the tag but the issue again arrives with the swimmers head going in and out of the water. This would be bringing the tag into and out of the read range field as far as the Reader is concerned and changing the tags state back very quickly rendering the approach of no use.
I tested this manually using the software, antenna and tag and each time I removed the tag from view it reset itself and was available again to read immediately.

An adjustment to this is potentially using some logic with software and manually ignoring any additional reads for a set period of time from the first tag read (that came into range going up the pool) for a period that guesses how long it would take a swimmer to be back down the pool out of sight, ignoring all other data in the mean time.

One obvious issue with this is the time set to be back out of sight could differ greatly between both swimming strokes and swimmer.

There is also a lot of data processing and filtering going on here just for an approximate position and lap count if it is scaled up even using the very low estimate numbers mentioned in the RSSI section above.

Having said that it does provide a way to acquire some tracking of the swimmer. Data processing can easily be scaled with cloud infrastructure. Some consideration would need to be given to the transmitting of data also to the cloud via the local network from the pool if the volume was to increase considerably.

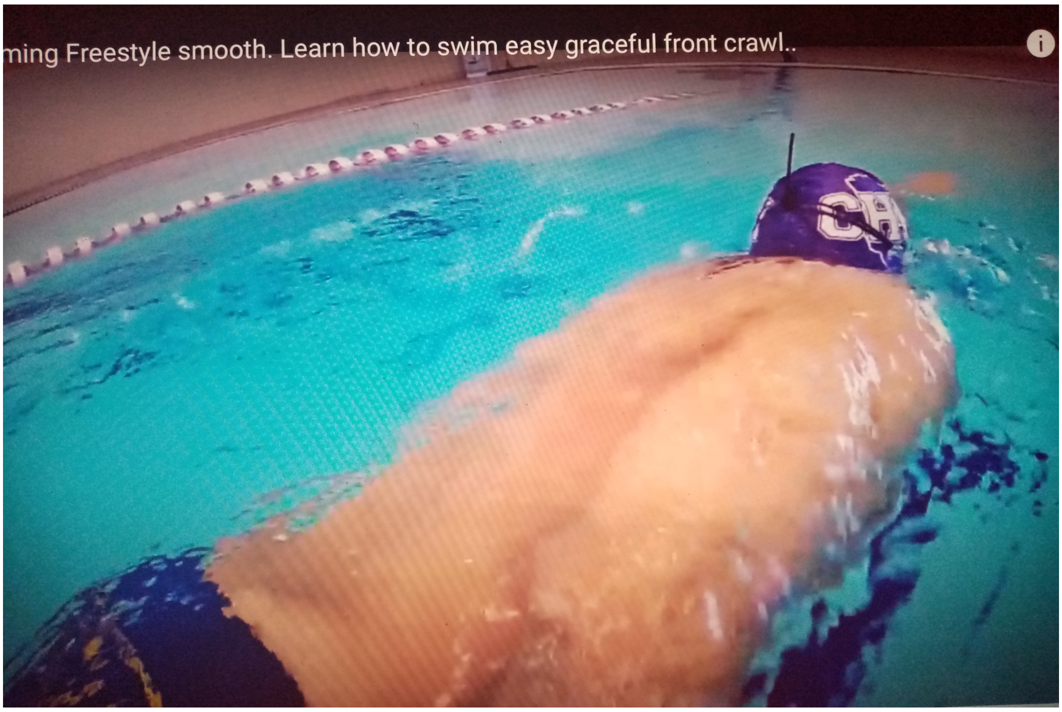
Proposed Solution

While each of the previous approaches has suffered from the same issues of tag angles, unreliable RSSI values, high volume of data processing and not knowing the position of the swimmer reliably, another approach that exists could offer a neater solution to address some of these issues.

As with all these approaches it is theory until practically verified in the pool itself. It is based on the front-crawl but may work for both the breast-stroke and butter-fly. How this could work for backstroke has not yet been determined.

"BACK OF THE HEAD" APPROACH

Position the tag on the back of the swimmers cap. See head position of an elite swimmer below during the front-crawl.



By using this position, when the swimmer pushes off from the wall the first time (on the side with the Antenna) the tag should be picked up. Diving starts from race blocks are not being considered for this.
Now we have an approximate position of the swimmer that also relates to the starting point within a much shorter range of accuracy.

Set the tag session to "*Single Target with Supression*" along with a Read Mode called "*Dense Reader M8*" which allows for the deepest scan of the tag and is the most interference tolerant setting by the Impinj Reader.

This will pick up the first read of the tag and suppress it. This will be picked up again within the read range as the swimmer head goes in and out of the water but with one potentially big difference.
Once a tag state suppression time can be manually set to a point that will get the swimmer down the pool out of range then on return due to the tag being positioned on the back of the head there should be no reads at all until the swimmer approaches the wall and either does a tumble turn which should flash the back of the head to the antenna (based on viewing slow motion footage of swimmers this appears to be the case) and also in the even of a non-tumble turn at the wall then the back of the head will be picked up again.

This would then register a completed 50m distance (approximate) and a lap count also along with a more accurate time.

This would also remove the problem of the manual suppression time covering all strokes as it only has to cover the swimmer going down the pool as the tag positioned on the back of the head should be naturally hidden all the way back.

This also greatly reduces the amount of data processing especially when scaled up as the tag just needs one read at the start and can just ignore all the data for a certain time period (e.g. 10 seconds) and then start listening again for that particular tag. So for each tag only one piece of data is needed per lap to transfer to the cloud as opposed to potentially hundreds per tag per lap.

As the data processing would be greatly reduced so would be the complexity of the cloud application architecture.

This could be an effective solution (if proven like all others in the water) for the front-crawl, breast-stroke and butter-fly.

The backstroke due the position of the head in the water would appear to be an issue with this approach. Even if the tag was not on the back of the head, the swimmer's head is facing away from the antenna at the start anyway and only facing the antenna on the approach back up the lane.
Most backstroke swimmers turn around to tumble at the wall so this would potentially be picked up at that point like all the other strokes but would essentially be missing 2 lengths before it started.

Known Blockers

Currently none

Next Steps

Resume implementation of the standalone Java application to interface with the Impinj Reader and start putting in place the chosen software solution.

The design of this standalone application can still be done in a way to “swap in/out” solutions using something like a *Strategy Design Pattern*, which would make it easier for adjustments or additions at a later point.