LSencom

Making LoRa Sensing Coexist with Communication

Paper Review by Raheem Idowu

10/27/25

LSencom

(a) A simple deployment

Enabling LoRa sensing alongside communication

LoRa

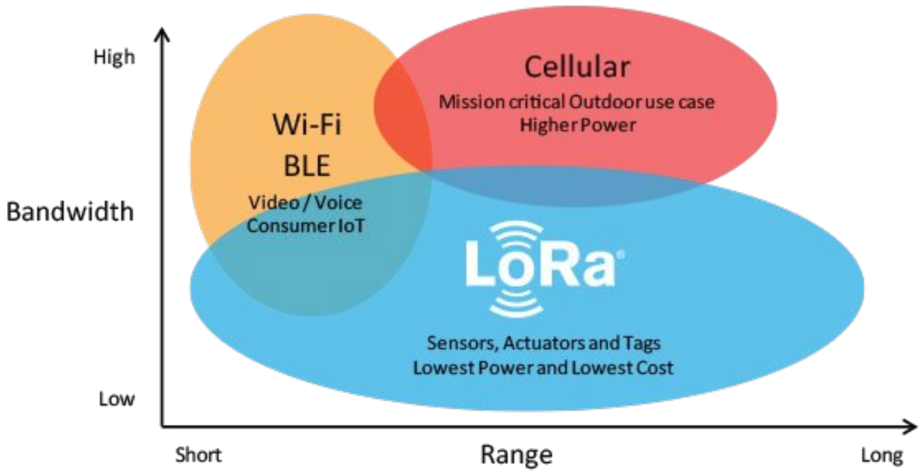
LoRa (Long Range) communication is:

- Low power
- Long range
- Low bitrate

Very useful for IoT sensors

Low power through high bandwidth

“Chirp spread spectrum”



LoRa Chirps

LoRa PHY uses chirps

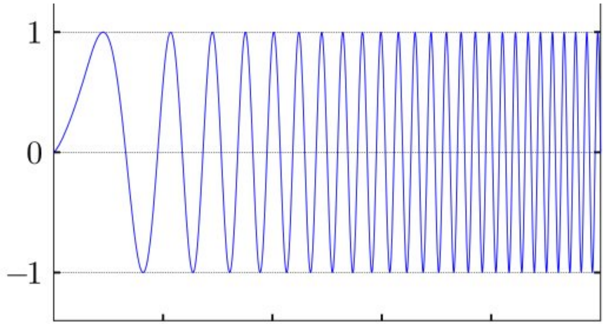
Signal that continuously changes frequency

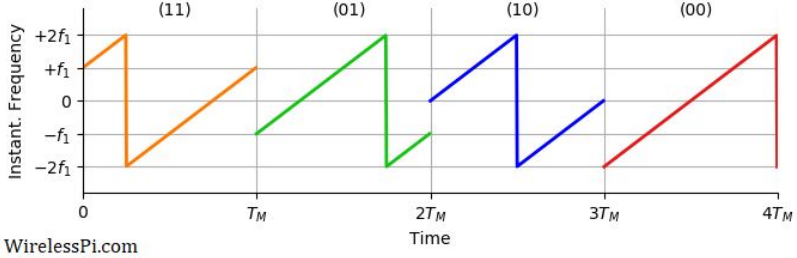
Uses a large bandwidth channel

Upchirp (and downchirp)

Data modulated by rotating the upchirp

Lora: 18 distinct upchirps





Sensing with LoRa

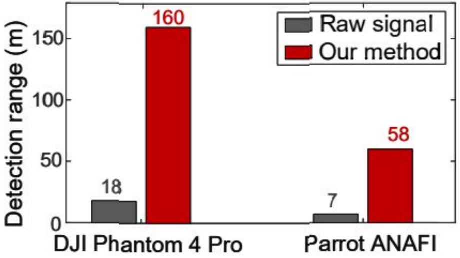
Why sense with LoRa?

Longer range sensing possible: 100m vs. 10m (WiFi)

LoRa frequency: 915 MHz (North America)

Much lower than WiFi, mmWave etc.

Low power = battery powered sensing?



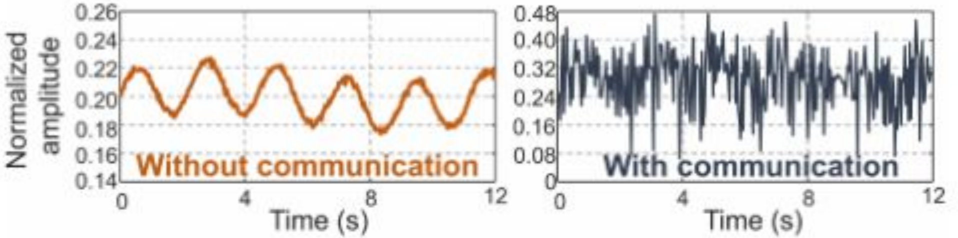
Boosting the Long Range Sensing Potential of LoRa

Binbin Xie, Minhao Cui, Deepak Ganesan, Xiangru Chen, Jie Xiong
University of Massachusetts Amherst

Challenges with LoRa sensing

Simultaneous communication causes interference

Why? Weak reflected signal & sensitive signal amplitude

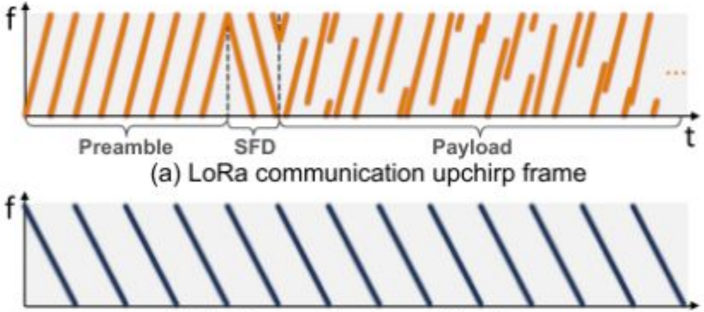


(b) Impact of communication interference on respiration sensing

Figure 1: Contact-free LoRa sensing cannot work in the presence of LoRa communication.

Unsatisfactory to frequency- or time-multiplex sensing and communication

Idea 1 - use downchirps



(a) LoRa communication upchirp frame

(b) LoRa sensing downchirp frame

Downchirp is orthogonal to upchirp

Should reduce interference

Requires configuration of COTS LoRa node

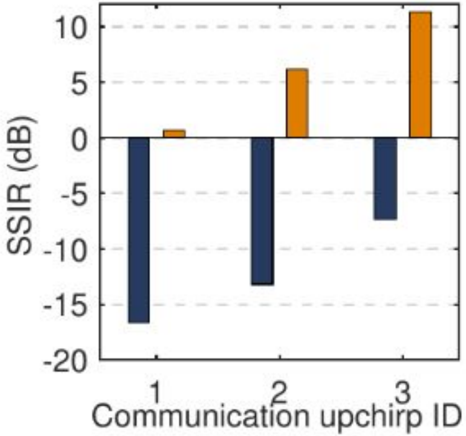
Evaluated using SSIR “Sensing signal to interference ratio”

P_(sen_tar) computed by comparing sensing signals with and without target

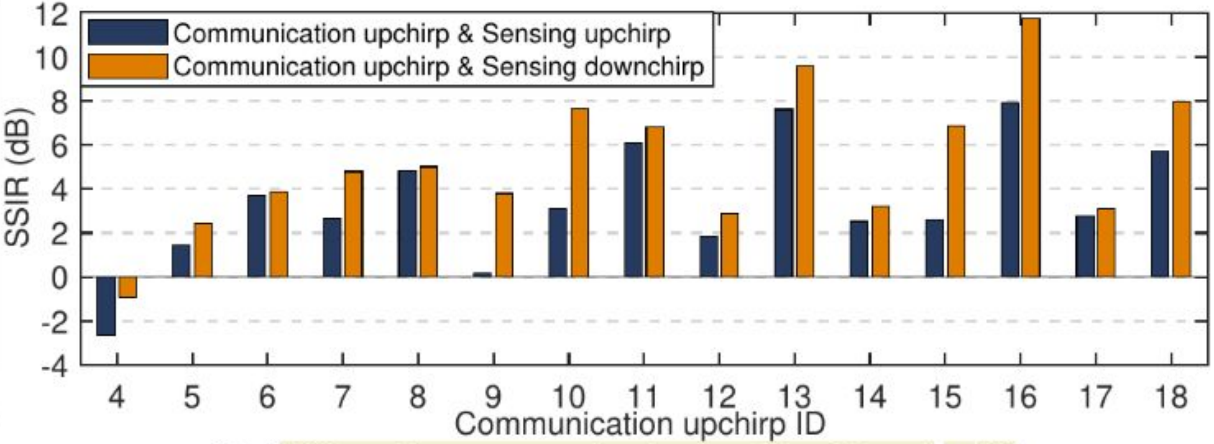
Isolates the dynamic components

$$SSIR = \frac{P_{sen\_tar}}{P_{com\_in}},$$

Idea 1 - Evaluation



(b) SSIR under communication upchirp ID = 1 ~ 3



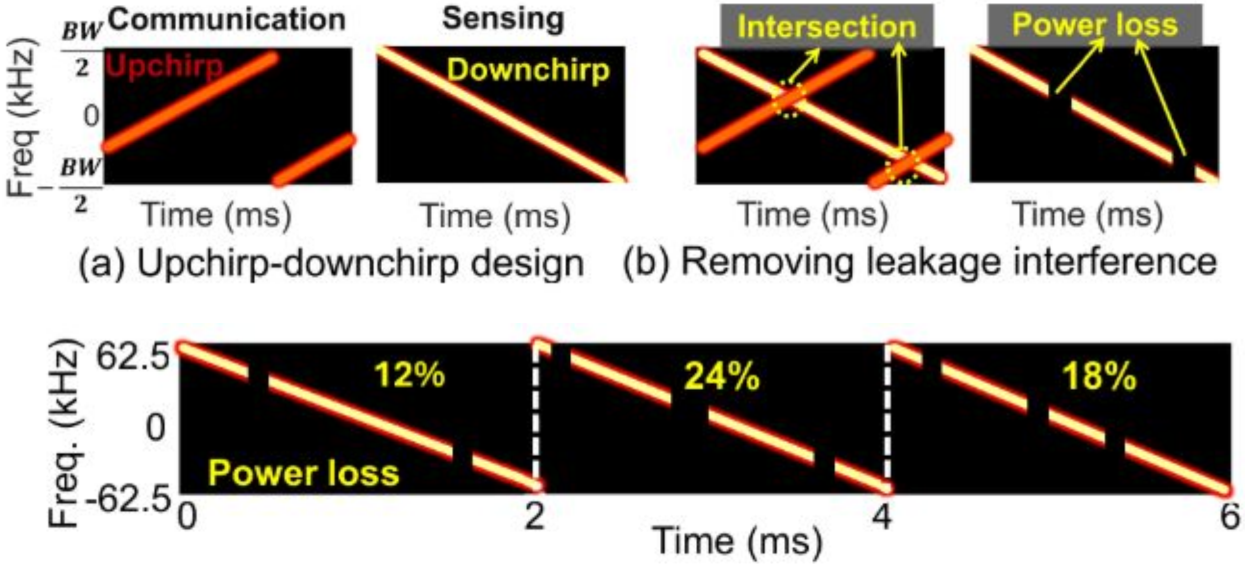
(c) SSIR under communication upchirp ID = 4 ~ 18

Communication & sensing node uses one fixed upchirp (or downchirp)

Idea 2 - Power leakage

Still leaks power to the downchirp! (intersection)

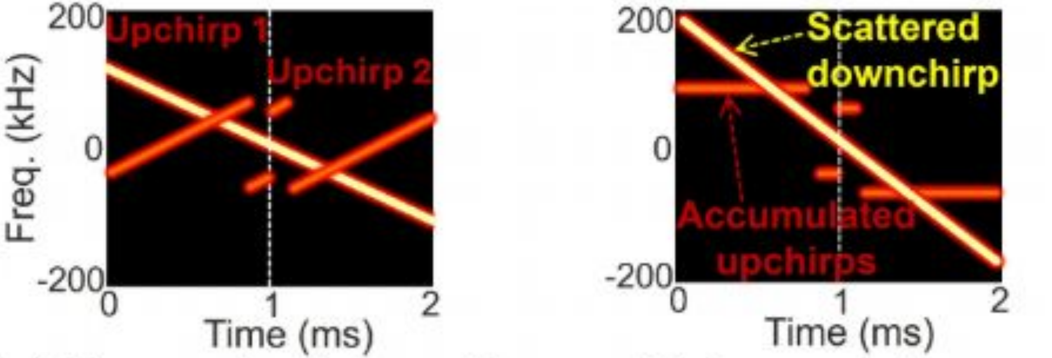
Remove them? Causes inconsistent amplitude = bad sensing!



Idea 2 - How to remove intersections?

Accumulate the upchirps (LoRa demodulation). Doesn't touch downchirp

Identify those frequencies, remove with bandstop filter



(a) Communication upchirp & sensing downchirp (b) Accumulating communication upchirps

Idea 3 - Chirp rotation

Any power variation = bad sensing

Solution: Novel chirp rotation

Rotating received downchirp 180 across the center (time-frequency plane)

One chirp = target information constant (short duration)

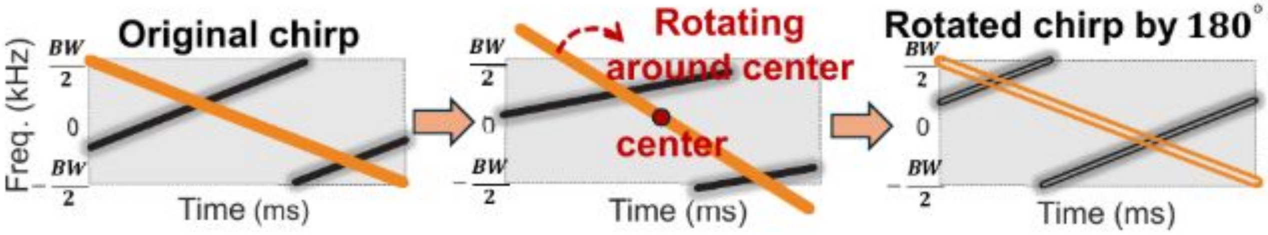
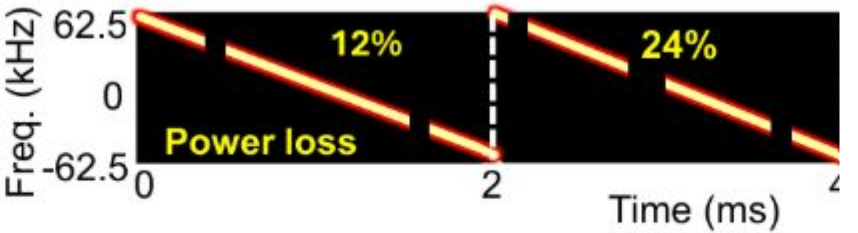


Figure 14: Chirps before and after rotation.

Idea 3 - How does chirp rotation help?

These two points are different:

- Center for downchirp (sensing node)
- Center for upchirp (communication nod



which means intersection point between:

- Original downchirp and rotated chirp
- Rotated downchirp and rotated upchirp

are different as well!

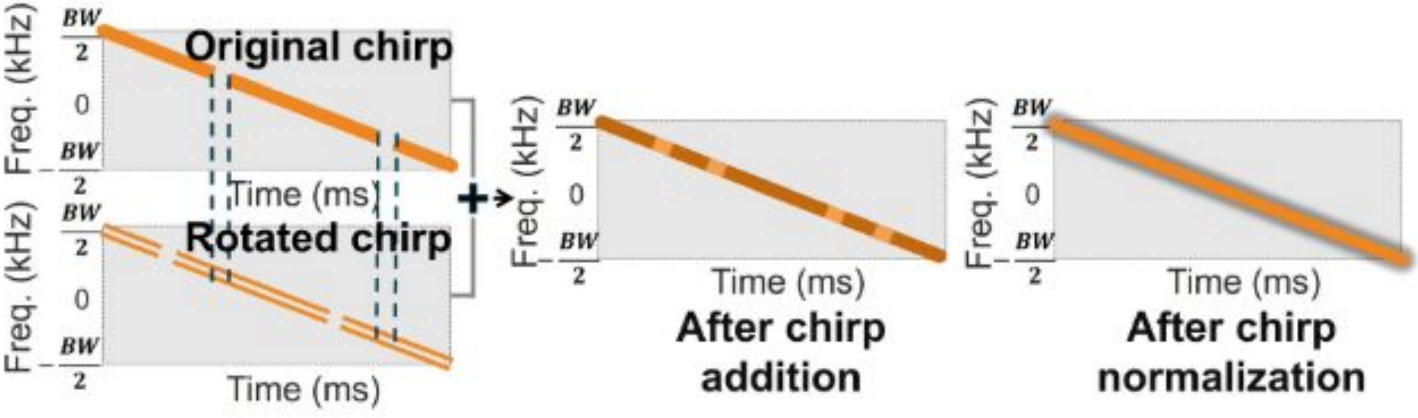
Everything else the same (amplitude, frequency, phase, target movement info.)

Idea 3 - Chirp rotation procedure

1) Receive combined sensing and communication signals

2) Demodulate communication, remove intersections

3) Rotate to get a new downchirp, add together and normalize



Idea 3 - Frequency synchronization issues

No time synchronization causes:

Central Frequency Offset (CFO) & Sampling Timing Offset (STO)

Rotated downchirp “ends up in the wrong place”

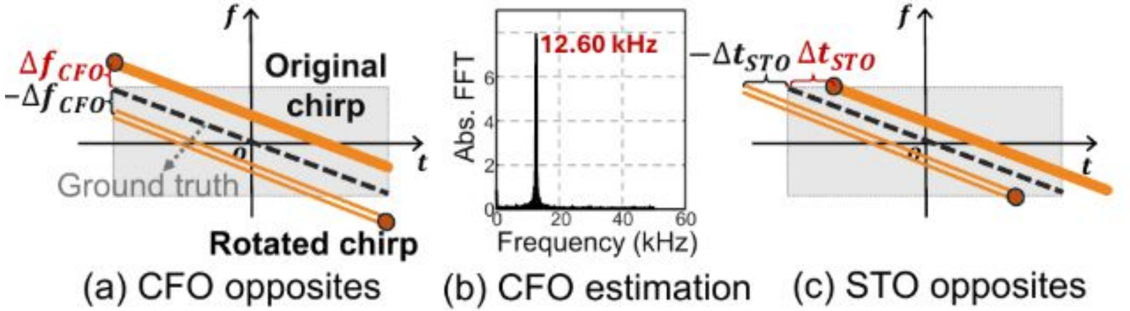


Figure 18: Addressing CFO and STO.

Idea 3 - Frequency synchronization fixes

Estimate CFO with symmetry (and math)

$$R(t,f) = S_{chirp}(t,f)e^{-j[2\pi\Delta f_{cfo}t+\phi_{c\_ini}]}(H_s + H_d),$$

$$R_{rot}(t,f) = S_{chirp}(t,f)e^{-j[2\pi(-\Delta f_{cfo})t+\phi_{c\_ini}]}(H_s + H_d).$$

$$R(t,f)/R_{rot}(t,f) = e^{-j2\pi(2\Delta f_{cfo})t}.$$

STO is removed by time alignment (thanks to identical frequency)

Idea 3 - Frequency synchronization fixes

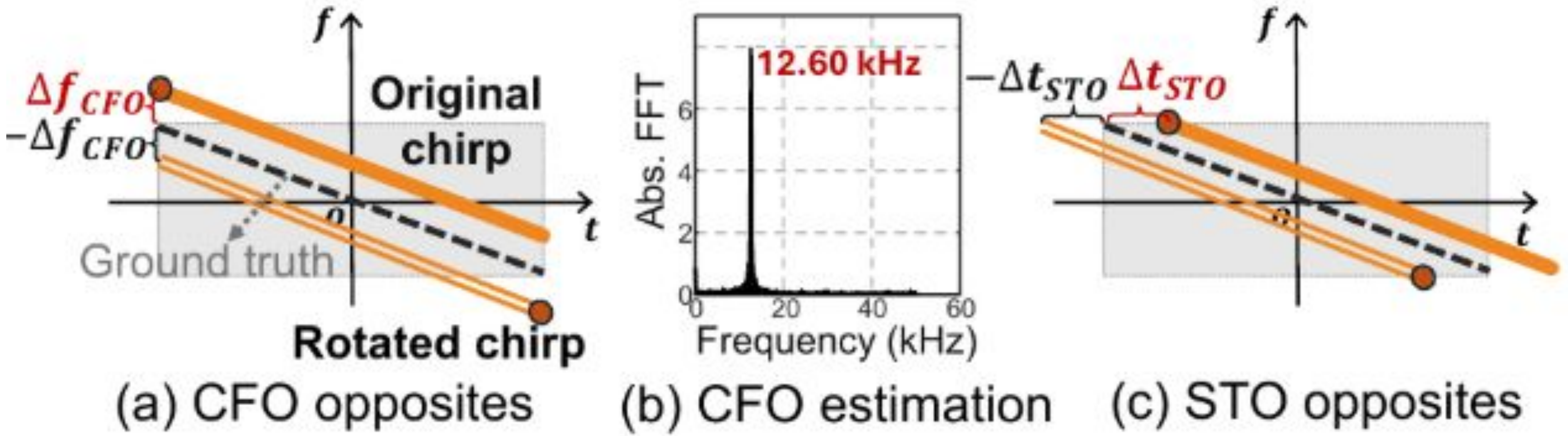


Figure 18: Addressing CFO and STO.

Implementation and experiment

LoRa gateway = USRP X310 (SDR)

LoRa sensing & communication node = Arduino Uno + Semtech SX1276

Experiments:

1) Baseline improvement
2) Multiple communication nodes
3) Real-world sensing
4) Sensing impact on communication

1) Baseline improvement (power ratio)

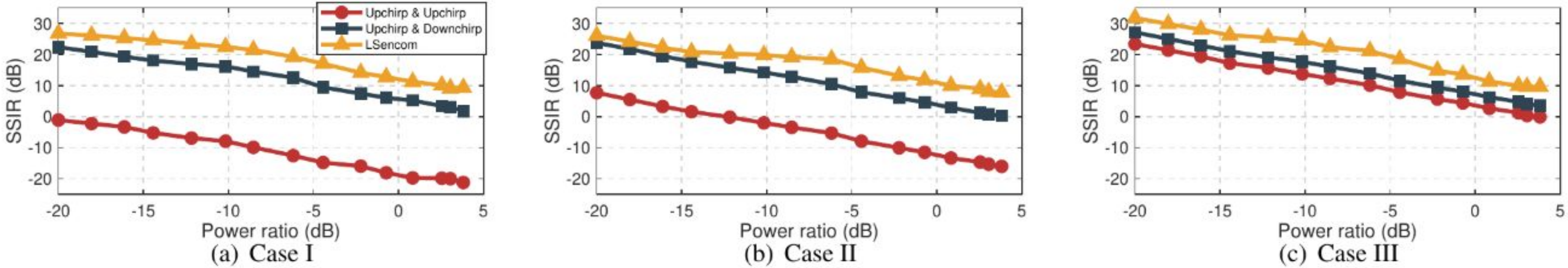


Figure 20: Impact of power ratio between received communication signal and sensing signal.

Case 1: Same parameters (SF, BW), same chirp ID

Case 2: Different parameters (SF, BW), same chirp ID

Case 3: Different parameters (SF, BW), different chirp IDs

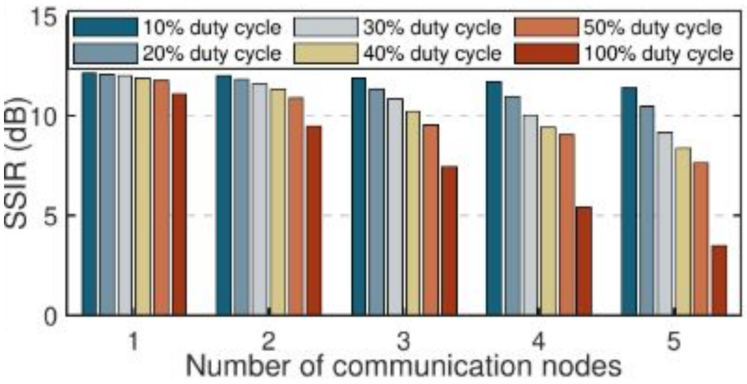
Takeaway: LSencom essentially mitigates all interference

2) Multiple nodes

1 sensing node & 1 - 5 communication nodes with different parameters

Duty cycle: how often you can send

Takeaway: LSencom resilient with multiple nodes with low duty cycles



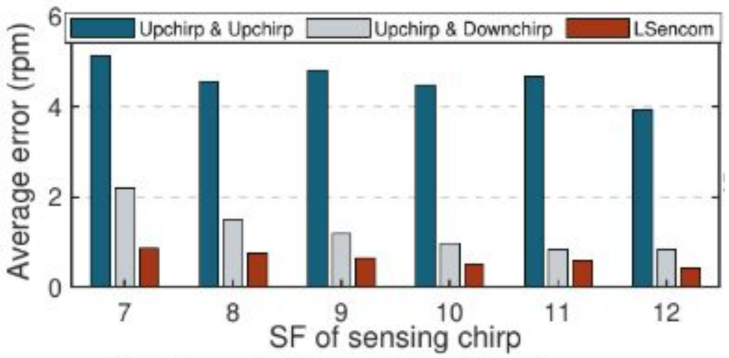
(b) SSIR with different duty cycles

3) Real-world sensing

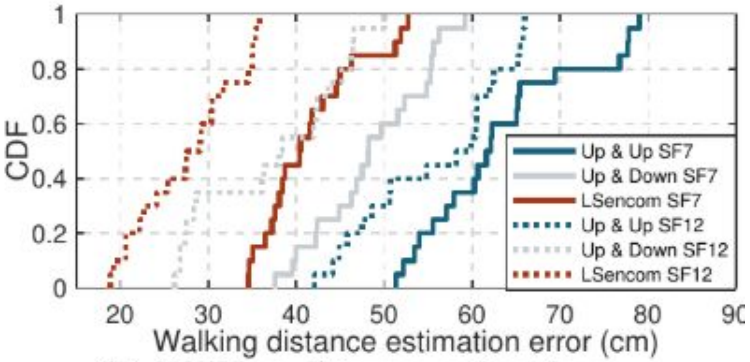
Respiration monitoring and walking sensing (NLoS)

Takeaway: Significantly reduces error compared to baseline

(Ground truth is Google Fit? Does that come with a monitor?)



(b) Respiration rate estimation



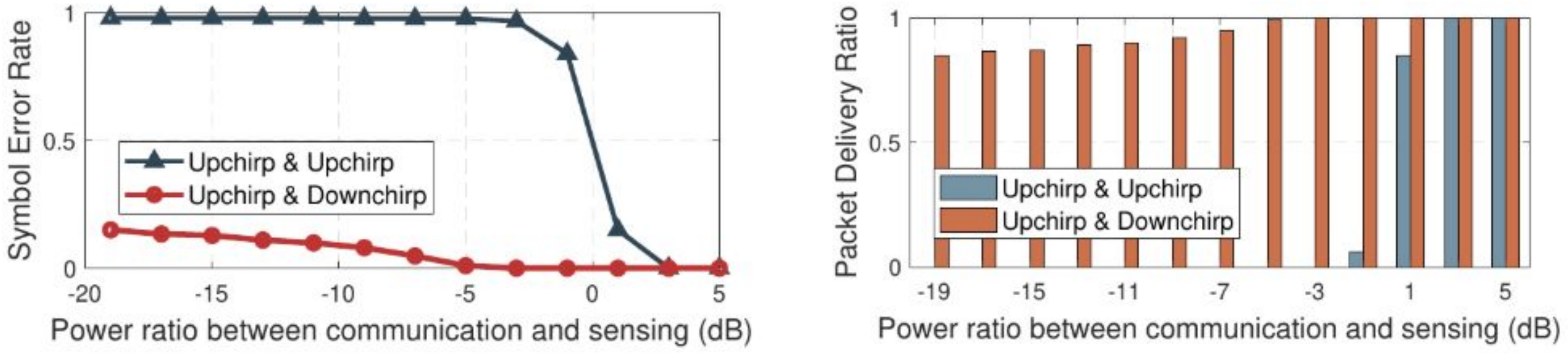
(b) Walking distance estimation

4) Impact on sensing on communication

LoRa synchronization uses downchirps = interference possible

Uses a method to avoid this called FlipLoRa

Takeaway: Upchirp-upchirp prevents almost all communication - use downchirp!



(a) Symbol error rate

(b) Packet delivery ratio

Discussion

Limitations / future work:

5 communication nodes too little

- LoRaWAN gateway can support hundreds to thousands devices

Sensing-to-sensing interference

Sensing and communication from same node

My takeaway:

Technically simple (no ML), good results. LoRa is cool!

Range not yet useful w/o multi-person sensing?

Thanks for listening!

Perusall time…